E-blocks go Twitter!

elektor

www.elektor.com

Practical, quick to implement, reusable & multi-platform
FT232R USB/Serial Bridge/BOB

+ ATM18/CMPS03 Compass

+ J²B ARM Cortex-M3 MMI

USB Long-Term Weather Logger

using I²C sensors for temperature, atmospheric pressure and humidity

R38
09

9 770268 451166

Sensitive to sensors

In all honesty there's often a gross discrepancy between the number of pages devoted to technology and projects of the "look-what-this-microcontroller-can-do-on-a-green-circuit-board" type and the slight coverage of sensors, those analogue components only a handful of people seem to be able to understand and work with these days. This month, we try to redress the balance.

At tech conferences, lots of raised hands in response to "hey guys, a workaround please to ARM7 TC/IP stack overflow error 5T26-i.822g but no prioritized interrupts and keep it all in Flash" but dead silence when the same crowd is prompted for wiring up a Kelvin junction or a word or two on resistor selection in a low-noise input amplifier.
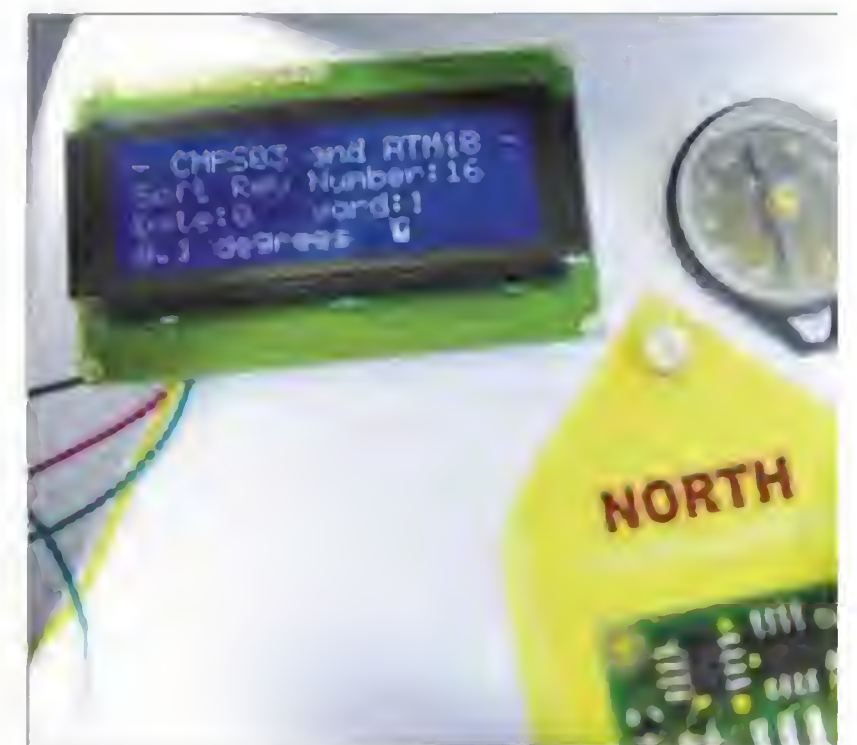
That is not to say the microcontroller and analogue fields are incompatible in any way, or mutually exclusive, it's just that that one needs the other to culminate in a working product. Fortunately sensor manufacturers are working hard to make their products as microcontroller-savvy as possible, while pizza-powered youngsters are providing code for their micros to get all the processing done as fast and user friendly as possible. With good results, as you can see from the USB Long Term Data Logger on page 16 and the ATM18 Compass on page 32. The Data Logger is accompanied by a separate article on I2C sensors on page 22 explaining some of the design backgrounds. Sensors want to be promoted, too. When I mentioned the use of a CMPS03 magnetic field sensor in our ATM18 Compass, Gerry at Devantech (Robot-Electronics) UK not only set up an exclusive reader offer but also sent me a sample of their latest compass sensor module.

There are other gems in this edition and they are in unexpected sections. Like the hardware for our DSP programming course that's attracting quite some attention worldwide, our new FT232/USB BOB (break out board) or, at returning to the all-analogue level, the Chaos Generator introduced on this month's Retronics pages. But your sensors may tell you differently.

Enjoy reading this edition,
Jan Buiting, Editor

CONTENTS

elektor | international media bv

Elektor International Media provides a multimedia and interactive platform for everyone interested in electronics. From professionals passionate about their work to enthusiasts with professional ambitions. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analogue and digital; practical and theoretical; software and hardware.

ANALOGUE • DIGITAL
MICROCONTROLLERS & EMBEDDED
AUDIO • TEST & MEASUREMENT

12-Watt PAR30 LED spotlights with dimming function

FZLED recently released a new line of 12-Watt PAR30 LED Spotlights that have lifetimes of more than 35,000 hours. Consumers can choose between a 'warm white' 3000K CCT or a 'cool white' 6000K CCT that provide luminous fluxes (lm) of 550 and 700, respectively. FZLED's commitment to R&D has enabled them to incorporate a creative dimming function into their PAR30 LED Spotlights.

The FZLED12-Watt PAR30 energy-efficient LED spotlights have an AC input voltage range of 90–264V and can be placed directly into standard E26/E27 sockets, hassle free. Furthermore, users can choose between a 55° or 120° beam angle. FZLED's built-in, dimming feature allows the spotlights to function at 12%, 25%, 50%, or 100% of light intensity. The desired level of brightness is easily selected by pressing the on/off power switch multiple times.

The FZLED 12-Watt PAR30 LED Spotlights are currently available in Taiwan and Singapore. FZLED is excited to build relationships with more distribution partners in order to provide consumers around the world with innovative, energy-saving, and high-performance lighting solutions.

www.fzled.com.tw   (110582-II)

Micro-inverter reference design board

Future Energy Solutions recently released and demonstrated a micro-inverter reference design board, providing a model for the rapid development of highly efficient and reliable micro-inverter end products.

The reference design is suitable for micro-inverters serving Photovoltaic (PV) solar panel arrays rated up to 200 watts and comprising up to 72 cells.

Developed in collaboration with Future Electronics' franchised suppliers Freescale Semiconductor and Fairchild Semiconductor, the design is a two-stage grid-connected micro-inverter providing high efficiency of up to 95% through the implementation of innovative design features including a sophisticated maximum power point tracking (MPPT) technique.

The system is comprised of two boards: a controller board which features the 16-bit MC56F8257 digital signal controller from Freescale, and an inverter board which includes the DC-DC boost and DC-AC inverter stages and an auxiliary power supply. Customers can use the boards as a development platform to which they can easily add peripheral features such as a display screen, user interface and communications. They can also source the original design files from Future Electronics and modify them freely.

Danny Miller, vice-president of Future Electronics' Future Energy Solutions division, which serves OEMs in the renewable and alternative energy markets, said: "Micro-inverters are a hot segment of the fast-growing market for solar inverters, which is expected to grow to $8.5 bn in sales in 2014. This reference board is a useful tool for our customers as they work to optimise their micro-inverter designs and adopt relevant new technology."

The design has adopted a non-isolating topology consisting of a DC-DC boost stage followed by a DC-AC inverter. Its omission of an isolating transformer, normally found in existing micro-inverter designs, helps to reduce losses markedly during power conversion.

The topology allows only a small 50 Hz ripple current to be reflected back from the 230 V AC load to the PV solar panel. The ripple current and ripple voltage are used to implement a fast MPPT technique called ripple correlation control, which is an effective means for capturing the maximum possible power from the PV solar panel throughout the hours of daylight.

Improvements over conventional micro-inverter designs have addressed durability issues as well as efficacy. The micro-inverter was designed from the outset to achieve long-term reliability both through significant derating of components, and through avoiding the use of life-limited aluminium electrolytic capacitors.

Future Electronics provides the developer with two alternative implementations. The demonstration at InterSolar is of an all-analogue implementation, which uses active components from Fairchild, including FCB20N60 ultra-fast switch-off MOSFETs and FAN7393 gate drivers.

On its general release, the reference design will also provide a parallel topology in which the boost stage and inverter stage control and MPPT are implemented in software on the MC56F8257 DSC.

This software-based version allows the designer to take advantage of the many peripheral features integrated into the MC56F8257, which include multiple high-resolution PWM channels, two 8-channel 12-bit ADCs, and support for an OLED display. It also provides the ability to implement quickly new and improved MPPT algorithms — an important consideration given the rapid evolution that the PV inverter market is experiencing.

A standard 160-way Future-Blox interface connects the control board to the inverter stage.

www.futureelectronics.com   (110582-I)

New evaluation kit for Sensirion's differential pressure sensor

EK-P3, the new evaluation kit from Sensirion, represents a straightforward and cost effective option for testing the digital differential pressure sensors of the SDP600 series.

The set consists of a USB stick that is connected to the SDP610 sensor by an adapter

Wireless sensor networks to measure radiation levels

The creation of Libelium's Radiation Sensor Board has been motivated by the nuclear disaster in Fukushima after the unfortunate earthquake and tsunami struck Japan. The company wants to help authorities and security forces to measure the levels of radiation of affected zones without compromising the life of the workers. For this reason Libelium have created an autonomous battery powered Geiger Counter capable of reading radiation levels automatically and sending the information in real time using wireless technologies like ZigBee and GPRS.

The design of the sensor board is open hardware and the source code is released under GPL.

The idea is simple, each node acts as an autonomous and wireless Geiger Counter, measuring the number of counts per minute detected by the Geiger tube and send this value using ZigBee and GPRS protocols to the control point. The system is powered with high-load internal batteries what ensures a lifetime of years.

Using this technology radiation measurements can be taken in real time without compromising the life of the security corps members as they do not have to be inside the security perimeter in order to activate the Geiger counters. The information is extracted automatically and sent wirelessly to the Gateway of the network.

The Geiger tube integrated in the Radiation Sensor Board is sensible to Beta and Gamma particles as they can be detected omnidirectionally. Consequently the orientation of the Geiger

sensor with respect to the source of radioactivity is uncritical — only the distance matters. For this reason fitting the nodes in the right places is essential to detecting a possible leakage from a nuclear source.

www.libelium.com   (110582-III)

---

cable. With the help of the software, which is available online for download, the differential pressure sensor can be tested under realistic conditions by following five simple installation steps. Consequently, there is no need to program a microprocessor as

the evaluation kit maybe connected directly to a PC. The included software allows displaying measured values on the screen and additionally provides the option of exporting the data to an Excel spreadsheet. This enables the data to be saved and processed in a simple manner.

The differential pressure sensors of the SDP600 family, which has recorded millions of sales, have a digital (I²C) output signal, are extremely long term stable and impress with their excellent accuracy and sensitivity, even at very low differential pressure values. The high performance is reached due to the thermal flow through principle. Thanks to the new EK-P3, the customer can learn more about the differential pressure sensor, become convinced of the benefits it provides and experience these clear advantages in a highly time efficient and economical manner.

www.sensirion.com/datasheet_ekp3   (110582-IV)

New Xilinx FPGA and FTDI USB high-speed 2.0 module

DLP Design, Inc.'s new DLP-HS-FPGA2 is a high-speed FPGA module based on silicon from Xilinx and FTDI. This new version has a larger FPGA but is otherwise identical to the DLP-HS-FPGA. The DLP-HS-FPGA2 uses an

XC3S400A-4FT256C from Xilinx. It has the same high-speed USB 2.0 interface based on the FTDI FT2232H, and 32 M x 8 DDR2 SDRAM from Micron, as the previous version. The module also comes with a working reference design, and is available from both Digi-Key and Mouser.

The DLP-HS-FPGA2 module is a low-cost, compact prototyping tool that can be used for rapid proof of concept or within educational environments. A 10,000-line reference design is provided for the Spartan™ 3A FPGA on the DLP-HS-FPGA2 to those who purchase the module. The design was written in VHDL and built using the free Xilinx ISE™ WebPACK™ tools.

As a bonus feature, the second channel of the dual-channel USB interface is used to load user bit files directly to the SPI Flash. No external programmer is required. This

represents a savings of more than $200. All that is needed to load bit files to the FPGA on the DLP-HS-FPGA2 is a Windows software utility (free with purchase), a Windows PC, and a USB cable. The new product is priced at $179.95.

www.dlpdesign.com   (110582-VII)

Industrial versions of Elnec programmers

Elnec, Europe's leading provider of solutions for programming NAND Flash memory, microcontrollers and other programmable

devices, have released programmers type BeeHive204AP and BeeProg2AP, which are industrial versions of the BeeHive204 and BeeProg2 programmers respectively. The

new programmers now available on market were specifically developed for implementation into 3rd party automated programmers and automatic test equipments (ATE).

chipKIT: first Arduino™-compatible 32-bit microcontroller development platform

Microchip announces the launch of the first 32-bit-microcontroller-based, open-source development platform that is compatible with Arduino™ hardware and software. Designed and manufactured by Digilent, a Microchip Authorized Design Partner, the chipKIT™ platform is the first and only 32-bit Arduino solution to enable hobbyists and academics to easily, and inexpensively, integrate electronics into their projects, even if they do not have an electronic-engineering background.

The chipKIT boards and software provide more features, performance and functionality than any other Arduino solution on the market. With boards starting at just $26.95 each, academics and hobbyists can experience four times the performance of any existing Arduino solution and have projects up and running in minutes.

The platform consists of two PIC32-based development boards and open-source software that is fully compatible with the Arduino programming language and development environment. The chipKIT hardware is compatible with existing Arduino shields and applications, and can be developed using the Arduino IDE and existing resources, such as code examples, libraries, references and tutorials. The easy-to-use, low-cost solution supports project development by hobbyists and academics from many disciplines, such as mechanical engineering, computer science and even art. The PIC32-based chipKIT boards enable 80 MHz performance, and provide up to 512 KB Flash, with up to 128 KB RAM. They feature connectivity peripherals, including Ethernet, CAN, and USB (Full-Speed Host, Device and OTG), plus peripherals such as multiple timers, a 16-channel 1 MSPS Analogue-to-Digital Converter (ADC), two comparators, and multiple I²C™, SPI, and UART interfaces. The chipKIT integrates Microchip's PIC32 microcontroller which is the highest performance 32-bit microcontroller in its class, featuring the industry-leading MIPS32® M4K® core from MIPS Technologies, Inc.

The software has been engineered to ensure maximum compatibility with existing Arduino shields, applications and courseware. The Arduino programming environment has been modified and extended so that it supports the PIC32-based chipKIT boards, as

well as traditional Arduino boards. The Arduino standard libraries have been also been modified to support chipKIT boards and traditional Arduino boards. All of this work has been contributed back to the open-source Arduino community. With the exception of a small number of shields that require 5V operation, the vast majority of existing Arduino hardware and software applications are fully compatible with the chipKIT platform, without modification.

The chipKIT Uno32™ (part # TDGL002) development board, priced at $26.95, is a clone of the Arduino Uno board and features 128 KB Flash programme memory and 16 KB RAM, with two each I²C, SPI and UART peripherals. The chipKIT Max32™ (part # TDGL003) development board, priced at $49.50, is a clone of the Arduino Mega board and features 512 KB Flash programme memory and 128 KB RAM, with USB, CAN and Ethernet communication, as well as five I²C, four SPI, and six UART peripherals.

Both chipKIT boards and supporting open-source are available today, as well as chipKIT Network and I/O Shields.

www.microchip.com/get/TDD2   (110582-X)

The new programmers have several enhancements which will be appreciated by electronics manufacturers and programming centres. In particular, the dimensions of the programmers were reduced, the cases are more robust. Also, the programming modules now have a different construction to ensure greater mechanical stability. For easy and convenient connection of the programmers to 3rd party automated machines, a simple Elnec remote control application was enhanced.

Elnec continues its tradition of manufacturing high quality products and provides a worldwide unique 3-year warranty with this new series of programmers dedicated for industrial use. Updates to the programmer's software, including new device support, are available from the Elnec website free of charge. Elnec provides very flexible support and releases new software on average every two working days! Willing to consider requirements of manufacturers, Elnec is providing considerable quantity discounts on programming modules.

www.elnec.com

(110582-V)

Parallax: Li-ion Power Pack / Charger – 2 Cell

The Li-ion Power Pack-Charger – 2 Cell from Parallax is an integrated storage cell and charging system on a single 3" x 4" printed circuit board. It's compatible with most 18650-size Li-ion cells. Parallax offer High-Capacity Li-ion Cells (#28987) in their store that are compatible with this charger.

The new product features PCB-mounted cell holders with on-board charging circuitry, multiple power input/output options, on-board output fuse protection, nominal 7.4 VDC output; 8.2 VDC maximum, standard 3" x 4" PCB footprint which integrates well with the Board of Education® (#28150), Propeller™ Proto Board (#32212), or any application needing a reliable power supply with an integrated charging system. The charge/discharge switching circuitry is automatic.

The small board holds two rechargeable 3.7 volt Li-ion 18650-size cells, with multiple LED indicators providing charge readiness information for each individual cell. A status key for the LED indicators is on printed on the board. Aggressive holders retain the cells in any board orientation and in moderate shock environments, such as mobile robotic applications. Cells are not permanent, and are easily replaced.

The dedicated circuitry on the board continuously monitors the charging process to ensure safety, efficiency, and to maximize the number of charge/discharge cycles of each cell.

The new product is priced at $49.99.

www.parallax.com (search '28986')

(110582-VI)

austriamicrosystems: 100% carbon neutral status by 2015

austriamicrosystems revealed ambitious plans to reach carbon neutral status by 2015 and become the first semiconductor manufacturer worldwide to do so. In its aggressive, ongoing efforts to be environmentally responsible, austriamicrosystems has been actively reducing its carbon footprint since 2004, achieving a reduction of 50% of CO_2 equivalents or 31,000 tons until 2010 since implementing actions. Over the last two years austriamicrosystems has completely mapped the CO_2 generation of all company activities including its employees. In 2011, the company will further reduce CO_2 emission equivalents by more than 9,000 tons by switching to 100% green electricity based mainly on water generated sources.

John Heugle, austriamicrosystems' CEO, stated, "At austriamicrosystems we consistently review the impact of our business on the environment and take steps to reduce pollution on the planet. Focused on clean technology to reduce our CO_2 impact, we are proud to announce that we are making significant progress towards the zero carbon footprint goal we have set for the company. Investments into energy savings, sustainable energy projects and CO_2 reduction programs not only help our environment, but also provide significant economic advantages to us which create benefits to our customers. austriamicrosystems has taken a leadership role in this important effort and hopes other companies in the semiconductor industry will follow."

austriamicrosystems continuously invests in the efficient use of energy and natural resources for environmental excellence. Programs in operation or in development include thermo solar cooling, state-of-the-art cleaning of waste water and exhaust air, creating products for renewable energy

applications, and biomass plant construction to enable the use of renewable fuel sources for heating and chilling requirements. The implementation of these programs has already reduced the company's use of coal and oil generated electricity and will soon free austriamicrosystems from dependency on imported natural gas as a key step towards the carbon neutral target.

www.austriamicrosystems.com/environment

(110582-VIII)

Anritsu redefines broadband VNA market

Anritsu Company introduces the ME7838A broadband vector network analyzer (VNA) system that provides single-sweep coverage from 70 kHz to 110 GHz with operation from 40 kHz to 125 GHz, and utilizes an advanced design that eliminates the need for large, heavy millimetre wave (mmWave) modules and coax combiners. The ME7838A provides engineers, designers, and researchers with a system that conducts highly accurate and efficient broadband device characterization of active and passive microwave/mmWave devices, including those designed into emerging 60 GHz wireless personal area networks (WPANs), 40 Gbps and higher optical networks, 77 and 94 GHz automotive radar, digital radio links, 94 GHz imaging mmWave radar, and Ka-Band satellite communications.

The ME7838A is also well suited for conducting signal integrity measurements on emerging high-speed designs, such as 28 Gbps serializer/deserializer (SerDes) transceivers used on servers, routers and other networking, computing and storage prod-

ucts. The ME7838A, equipped with the 3743A mmWave module, can accurately measure 28 Gbps SerDes transceivers at the higher frequencies required for proper analysis.

Among the many advantages of the ME7838A is improved RF performance, due to an industry first, real-time power leveling control that provides the best power accuracy and stability to power levels as low as -55 dBm. The approach employed in the ME7838A takes less time, is less tedious, and more accurate than the conventional method of adjusting power level in the millimetre band through the use of electronically controlled mechanical attenuators and power linearity correction tables. The VectorStar® broadband system provides an accurate and fast real-time method to sweep power for compression measurements. The result is that the ME7838A performs the most accurate gain compression measurements on high-frequency active devices in the industry.

With the ME7838A design, mmWave modules can be mounted close to or directly on the wafer probe. This advantage, as well as the fact that the ME7838A transitions at 54 GHz, gives the broadband VNA the widest dynamic range in its class – 107 dB at 110 GHz and 92 dB at 125 GHz.

The ME7838A is the first broadband VNA to provide good raw directivity throughout the entire frequency range, due to its innovative design and elimination of the MUX combiners used in traditional systems. Best-in-class raw performance allows the ME7838A to offer engineers and designers improved calibration and consistent measurement stability of 0.1 dB magnitude and 0.5° phase across the entire 70 kHz to 110 GHz frequency range over a 24-hour period. Measurement speed is 55 ms for 201 points at 10 kHz IF bandwidth, 10 times faster than comparable broadband VNA systems.

www.anritsu.com   (110582-XI)

DFM Now! revolutionizes PCB CAM software

Numerical Innovations' new product DFM Now!™ allows PCB designers and engineers to verify that their Gerber and Drill files are ready for PCB manufacturing. It also facilitates PCB Quotation and has many other high-end CAM features only found in software costing thousands of dollars. DFM

Now! is being offered completely *free* supported by advertising sponsorships (a new concept for the PCB Software Industry), and may be downloaded from the DFM website. DFM Now! Is a revolutionary product for the PCB industry as it offers PCB designers and engineers powerful CAM DFM features, only found on expensive software, absolutely free because it is advertiser supported. It is a 'first of its kind' product for PCB design professionals," states Simon Garrison of Numerical Innovations.

DFM Now! brings design verification into a comprehensive, accurate and easy-to-use package for the PCB design professional. Having the same power and feature-rich intelligence of NI's other products, and being powered by their popular FAB 3000 engine (which has an installed base of over 1000 unique companies), DFM Now! Provides:

- design verification to bridge the gap between design and manufacturing with 'True DFM';
- greater performance in the areas of speed, usability, and reliability with the most intelligent GUI in the industry.
- a economic value to its customers, saving design professionals time and money on their critical designs.

While most PCB Layout tools are very good at performing DRC (Design Rules Check), they rarely capture many common DFM (Design For Manufacturing) problems that are hidden in the physical artwork (i.e. Gerber, drill files). It's the artwork which is ultimately used to manufacture the boards and any hidden problems in artwork will affect the manufacturability and yield of boards. DFM Now! allows PCB designers & engineers to verify that their Gerber and Drill files are ready for PCB manufacturing using 'True DFM' technology.

www.numericalinnovations.com

(110582-XIII)

eC-Reflow-Mate
A new, professional SMT reflow oven with unique features

In the wake of the popular SMT reflow oven introduced by Elektor in late 2008, we now present a new SMT oven, developed in cooperation with EuroCircuits Belgium, that is even more precise, has more room for PCBs, and can even be operated from a PC. In short, it is a truly professional machine that deserves a place in every electronics lab or shop where SMD boards are assembled on a regular basis.

eC-reflow-mate specifications

Supply voltage:	230 V / 50 Hz only	Menu languages:	English, French, German, Dutch, Italian, Hungarian
Power:	3500 W		
Weight:	approx. 29 kg	Temperature range:	25 to 300 °C
Dimensions:	620 × 245 × 520 mm (W × H × D)	Maximum PCB size:	400 × 285 mm
Heating method:	Combined IR radiation and hot air	Temperature sensors:	2 internal and 1 external (included)
Operation:	Directly using menu buttons and LCD on oven	Special features:	
	Remotely using PC software and USB connection		

Special features:
- Optimal temperature distribution thanks to special IR lamps
- Drawer opens automatically at end of soldering process
- Glass front for easy viewing

The **eC-reflow-mate** is ideal for assembling prototypes and small production batches of PCBs with SMD components. This SMT oven has a very large heating compartment, which provides plenty of space for several PCBs. Two built-in sensors and IR lamps with non-linear profiles, specially developed for this machine, help keep the temperature inside the entire oven compartment very uniform and constant. An additional sensor can be connected separately to measure the surface temperature of a component or the PCB. The oven is supplied as standard with five preconfigured heating profiles, which can easily be adapted to your own wishes. The accompanying PC software allows you to monitor the temperature curves of all sensors precisely during the soldering process, and it enables you to modify existing temperature/time profiles or create new ones. This can be done very easily by using the mouse to move

PC interface

The eC-reflow-pilot software for the eC-reflow-mate, which is compatible with Windows XP, Windows Vista and Windows 7, enables full remote control of the oven over a USB link. The screen provides a simultaneous display of the selected (configured) temperature/time profile and the actual temperature/time profile inside the oven. The individual temperatures of all sensors in the oven (2 or 3) are displayed constantly.

Temperature/time profiles can be
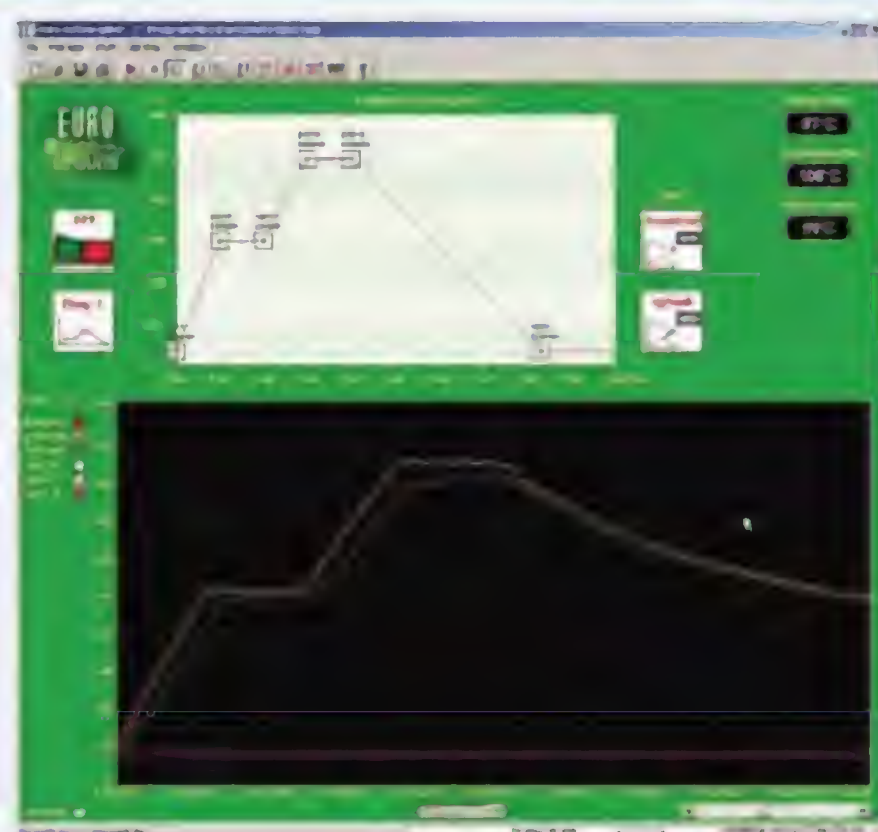
adjusted by using the mouse to drag one or more corner points of the curve in the horizontal or vertical direction to change the time or temperature, respectively. The screen also has buttons for saving new or modified temperature/time profiles or opening existing profiles. The oven can even be switched on or off from the PC. Switching to a different user interface language is easy. Five languages are presently available; more languages will be added in the future.

corner points on the screen in order to adjust times and temperatures. A glass front panel lets you keep an eye on what's happening inside the oven at all times.

The eC-reflow-mate features especially robust construction. The large drawer is mounted on multiple support rails, and it can be opened either mechanically or electronically. The drawer opens automatically at the end of the soldering process. A clever air circulation system in the oven maintains a uniform air temperature inside the entire compartment. The oven is well insulated to maintain the case temperature at a safe low level, even with prolonged use.

The eC-reflow-mate is an uncompromising SMT oven with extensive features, and it is a valuable asset for everyone who regularly needs to assemble PCBs with SMD components.

(100447-I)

The eC-reflow-mate is priced at € 2495 (plus VAT and shipping charges) and supplied directly by Eurocircuits in Belgium.

For ordering information, visit www.elektor.com/reflow-mate

Glass front for easy PCB viewing

Magnetic mounts for securing PCBs

Extra sensor for measuring the temperature of a component or the PCB

Specially designed IR lamps with non-linear heating profile

Sensor for temperature on top side of PCB

Sensor for temperature on bottom side of PCB

Large usable area with room for several PCBs

USB port for connection to a PC

Drawer supported by multiple rails opens automatically at the end of the soldering process

USB Long-Term Weather Logger

Using I²C sensors for atmospheric pressure, temperature and humidity

By Wilfried Wätzig (Germany)
wr-waetzig@t-online.de

This stand-alone data logger displays pressure, temperature and humidity readings generated by I²C-bus sensors on an LCD panel, and can run for six to eight weeks on three AA batteries. The stored readings can be read out over USB and plotted on a PC using gnuplot. Digital sensor modules keep the hardware simple and no calibration is required.
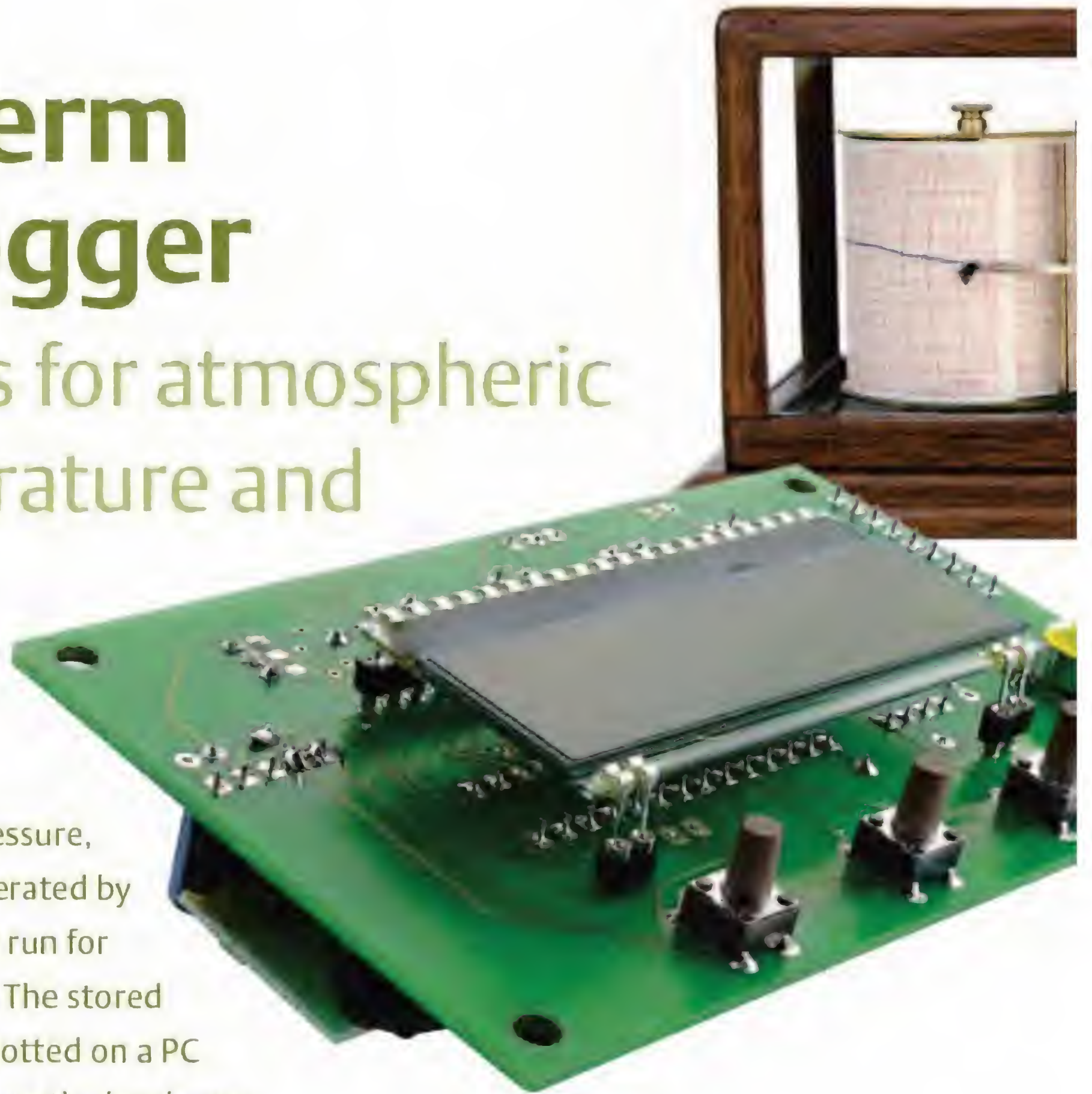
The author designed this data logger as a long-term recording system for pressure, temperature and humidity data. Samples are taken at regular intervals and stored in a serial EEPROM, which means that they are preserved even when power is lost. A serial-to-USB module allows the data to be read out by a connected PC for processing. The sensor modules (one for pressure and temperature, one for humidity) with I²C interfaces are described more thoroughly elsewhere in this issue and are supplied ready-calibrated [1], which simplifies construction and processing considerably. Thanks to its LCD panel and (rechargeable) battery power supply the logger can be installed anywhere for stand-alone operation. The use of a power-efficient ATmega88 microcontroller keeps the average current draw of the circuit below 2 mA.

The firmware is written in C and compiled using AVR-GCC. The source code is available as a free download, so that it can be modified to work with other types of I²C sensor if necessary.

Hardware

Since modules are used for some of the functional blocks, the circuit in **Figure 1** is simple and clear. At the heart of the circuit is the ATmega88 controller (IC2) [2], which is connected to the various modules. Two of these are connected in the usual fashion to port pins.

The DOGM162 display (LCD1) [3] is a two-line by sixteen-character unit without backlight. The display is driven in four-bit mode over PORTB. It was chosen because it can operate from a 3.3 V supply.

Mod1 is a new serial-to-USB converter module (BOB-FT232R) [4], which is also described elsewhere in this issue.

The other three modules (actually two modules and one IC) are connected to the microcontroller over an I²C bus: the Atmel incarnation of this is called a 'two wire interface' (TWI). This very widely-used bus allows up to 128 bus participants ('slaves') to be connected using just two signal wires. In this circuit the integrated TWI controller [5] in

---

Elektor Products & Services

- Printed circuit board: order code 100888-1
- Ready-programmed microcontroller: order code 100888-41
- BOB-FT232R V2.20 serial-to-USB module: order code 110533-91
- HH10D humidity sensor: order code 100888-71
- HP03S atmospheric pressure sensor: order code 100888-72
- Project software: file # 100888-11 (free download)

Items available via www.elektor.com/100888

Features

- Accurate measurement of atmospheric pressure, temperature and humidity without calibration
- Store up to six records (each containing pressure, temperature and humidity readings) per hour
- Non-volatile storage for 8191 data records
- Up to eight weeks' stand-alone operation from three AA cells
- Calibrated sensor modules with digital outputs for pressure, temperature and humidity
- Data can be read out over a USB interface

the ATmega88 is configured as the 'master' of the following slaves.

The HP03S pressure sensor from Hope Microelectronics (a company perhaps already known to *Elektor* readers through its radio modules [6]) contains a piezoresistive transducer and integrated 15-bit A/D converter (ADC), along with control logic and an I²C interface. The transducer outputs one voltage that depends on pressure and one that depends on temperature. These analogue values are alternately converted by the ADC and the results made available on the I²C interface. During the manufacturing process eleven sensor-specific calibration values with a length of two bytes are stored in the device's EEPROM, and these can also be retrieved by the microcontroller. A 32 kHz clock with an amplitude of 3 V is required to drive the ADC: since the 32 kHz oscillator on the ATmega88 has an output amplitude of only about 0.5 V, a BS170 (T1) is used for amplification.

The HH10D humidity sensor is made by the same manufacturer (Hope Microelec-
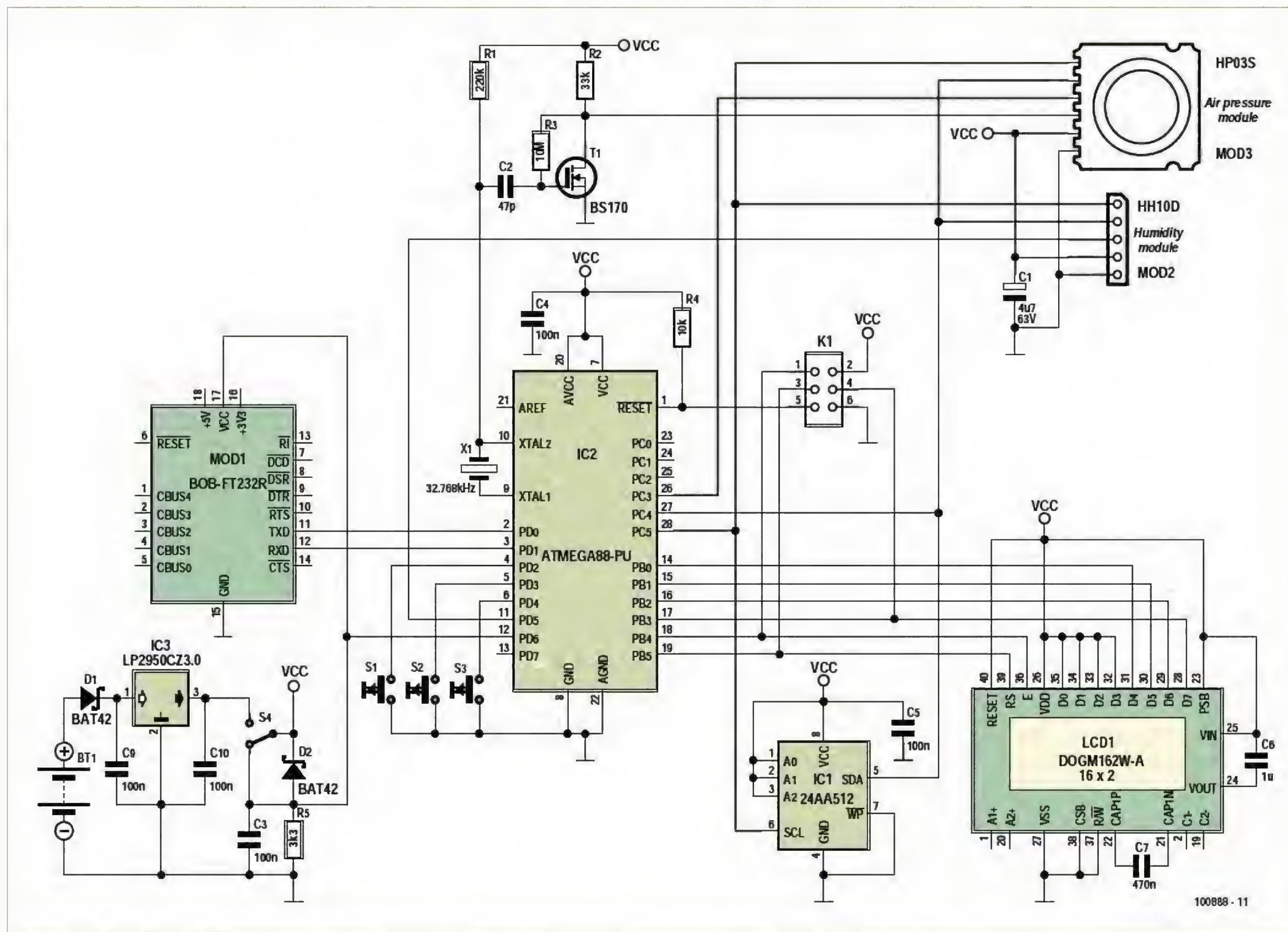
Figure 1. Circuit diagram of the logger: an ATmega88 and and I²C EEPROM are accompanied by four modules (USB interface, LCD, and sensors for humidity and for pressure and temperature).

COMPONENT LIST

Resistors
R1 = 220kΩ
R2 = 33kΩ
R3 = 10MΩ
R4 = 10kΩ
R5 = 1.3kΩ

Capacitors
C1 = 4.7µF 63V radial
C2 = 47pF
C3,C4,C5,C9,C10 = 100nF
C6 = 1µF MKT 5mm lead pitch
C7 = 470nF

Semiconductors
D1,D2 = BAT42
T1 = BS170
IC1 = 24AA512
IC2 = ATMEGA88-20PU, programmed, Elektor
 # 100888-41
IC3 = LP2950-3.3 or -3.0

Miscellaneous
S1,S2,S3 = 6mm switch, PCB mount
S4 = single-pole switch
X1 = 32.768-kHz quartz crystal
LCD1 = DOGM162W-A (Electronic Assembly)
Mod1 = BOB-FT232R-V2.20 (Elektor #
 110533-91)
Mod2 = humidity sensor HH10D (Hope RF,
 Elektor # 100888-71)
Mod3 = pressure sensor HP03S (Hope RF,
 Elektor # 100888-72)
K1 = 6-pin (2x3) pinheader (optional for ISP
 interface)
20-way socket strip SIL for LCD1
18-way (2x9) socket strip for MOD1
5-way socket strip for MOD2
IC socket for IC1 (8-way) and IC2 (28-way)
PCB, Elektor # 100888-1
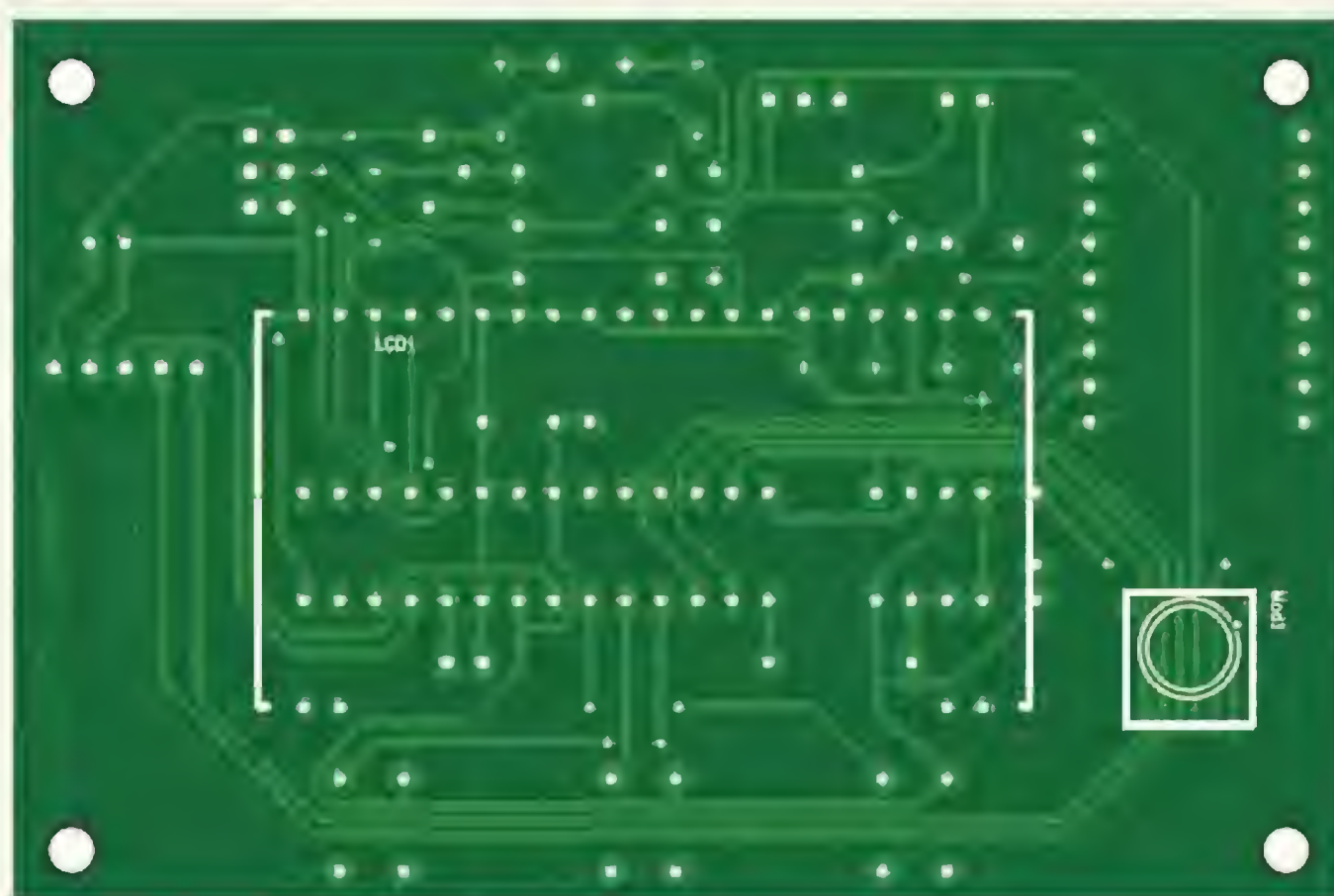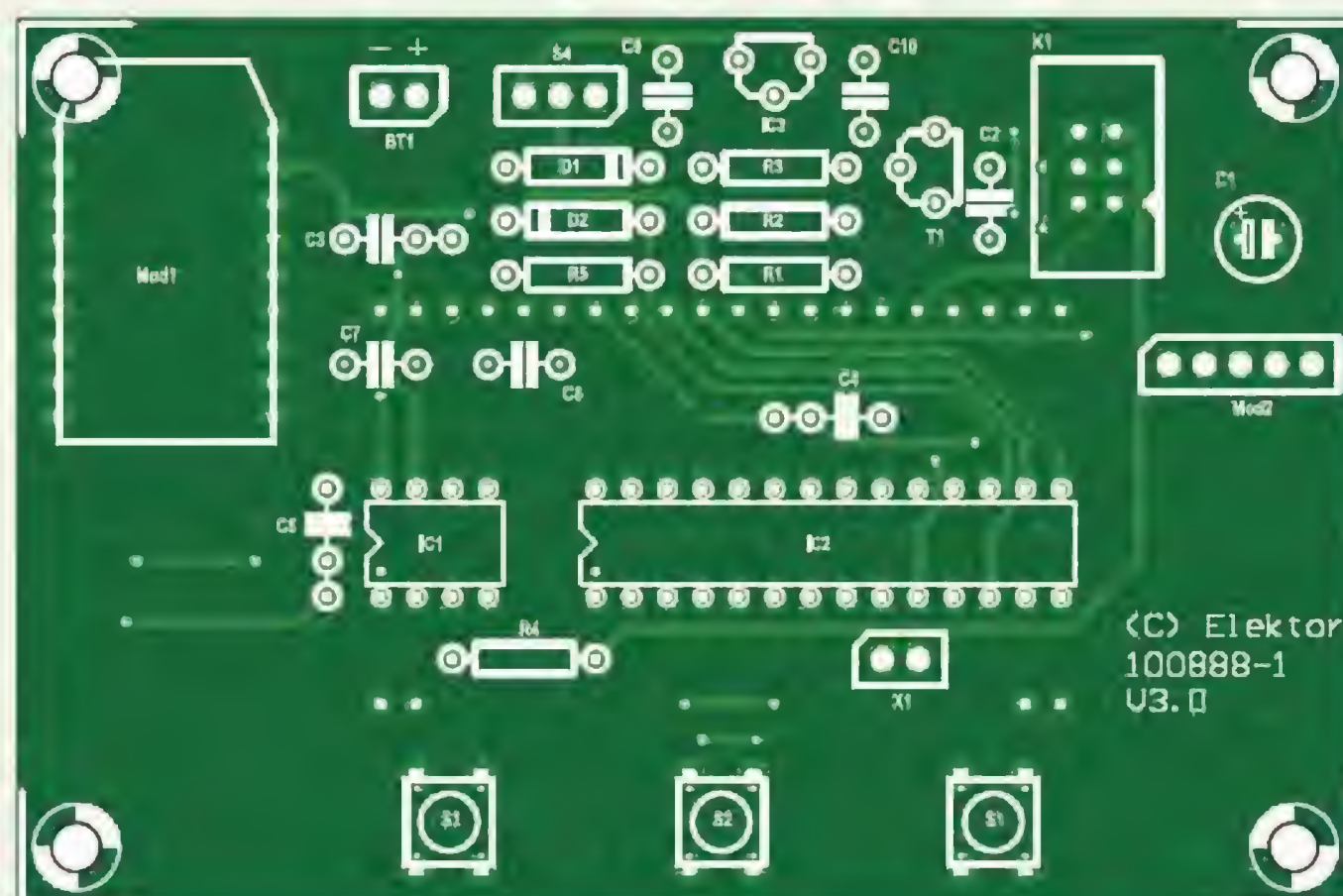
Figure 2. It is easy to populate the double-sided printed circuit board.

tronics). A humidity-sensitive capacitor is used as the transducer element, determining the frequency of an ICM7555 timer IC. The frequency, in the range 6 kHz to 7 kHz, is measured by the ATmega88 and then converted into a relative humidity value with the help of two calibration parameters again stored in a serial EEPROM in the module. The ATmega88 measures the frequency by counting the output pulses of the module over a one second period using the 16-bit counter TIMER1.

The I2C serial EEPROM type 24AA512P (IC1) [7] has a capacity of 64 Kbyte and stores the measured quantities (elapsed hours, time, humidity, temperature and pressure). Up to 8191 data records can be stored.

Three pushbuttons are provided for the user interface to the device. S4 is used to select between (rechargeable) battery and USB power for the unit, and K1 allows an AVRISP or compatible programmer to be connected.

Software
The firmware for the microcontroller is written in C and compiled using AVR-GCC 4.3.0 (WinAVR 20080610). Separate source files are used for the functions corresponding to each module.

The main program in `weather_station.c` calls as required routines in `lcd_driver.c` (to control the display), `usart_driver.c` (to control the serial interfaces via the USART) and `TWI_driver.c` (for I2C bus control). There are device-specific functions within each of these files.

TIMER2 is driven by the 32.768 kHz crystal and is configured to generate an interrupt once per second. The interrupt service routine `TIMER2_COMPA_vect` increments the current time (expressed in hours, minutes and seconds) and sets the event variable `flag1sec`. At the beginning of each minute a test is made to see whether a new set of readings is to be stored: this is done a preset number of times per hour. If new readings are required the event variable `flagmstor` is set.

These variables are checked in the infinite loop in the main program. If `flag1sec` is

set then the microcontroller is switched at the fifty-ninth second of the minute from SLEEP_MODE_POWER_SAVE to SLEEP_MODE_IDLE (which re-enables IO_CLK), so that TIMER1 can measure the frequency of the signal from the humidity sensor.

If the variable `flagmstor` is set then a reading is collected from the pressure sensor module, the calibration correction calculations are carried out, and the final results are written to the I2C EEPROM. The CPU then returns to SLEEP_MODE_POWER_SAVE and waits for the next interrupt from TIMER2.

Construction

The double-sided printed circuit board shown in **Figure 2** is, like the circuit diagram, very straightforwardly laid out. Apart from the pressure sensor all the components are leaded and are mounted normally. It is best to start with the SMD pressure sensor, which is not too tricky to solder by hand to the underside of the board. Also mounted on the underside of the board are the three buttons S1 to S3 and the sockets for mounting the LCD module. Sockets are soldered on the component side of the board for mounting the humidity module and the serial-to-USB module. It is a good idea to use sockets for the two ICs. The first version of the board made at the Elektor labs is shown in **Figure 3** and **Figure 4** (Figure 2 shows the final version).

The fuse settings which are to be programmed into the microcontroller using the ISP connector K1 are given in the text box. The programming job (and connector K1 itself) can be avoided if a ready-programmed microcontroller is used: this, along with the sensor modules and the serial-to-USB module, is available from the Elektor shop.

After carefully checking your soldering you can switch the logger on for the first time. Make sure S4 is in the correct position to select between external power (3.3 V at BT1) and power over USB. The unit requires no calibration.

Operation

The three buttons (S1, S2 and S3) allow the time and other parameters to be set.
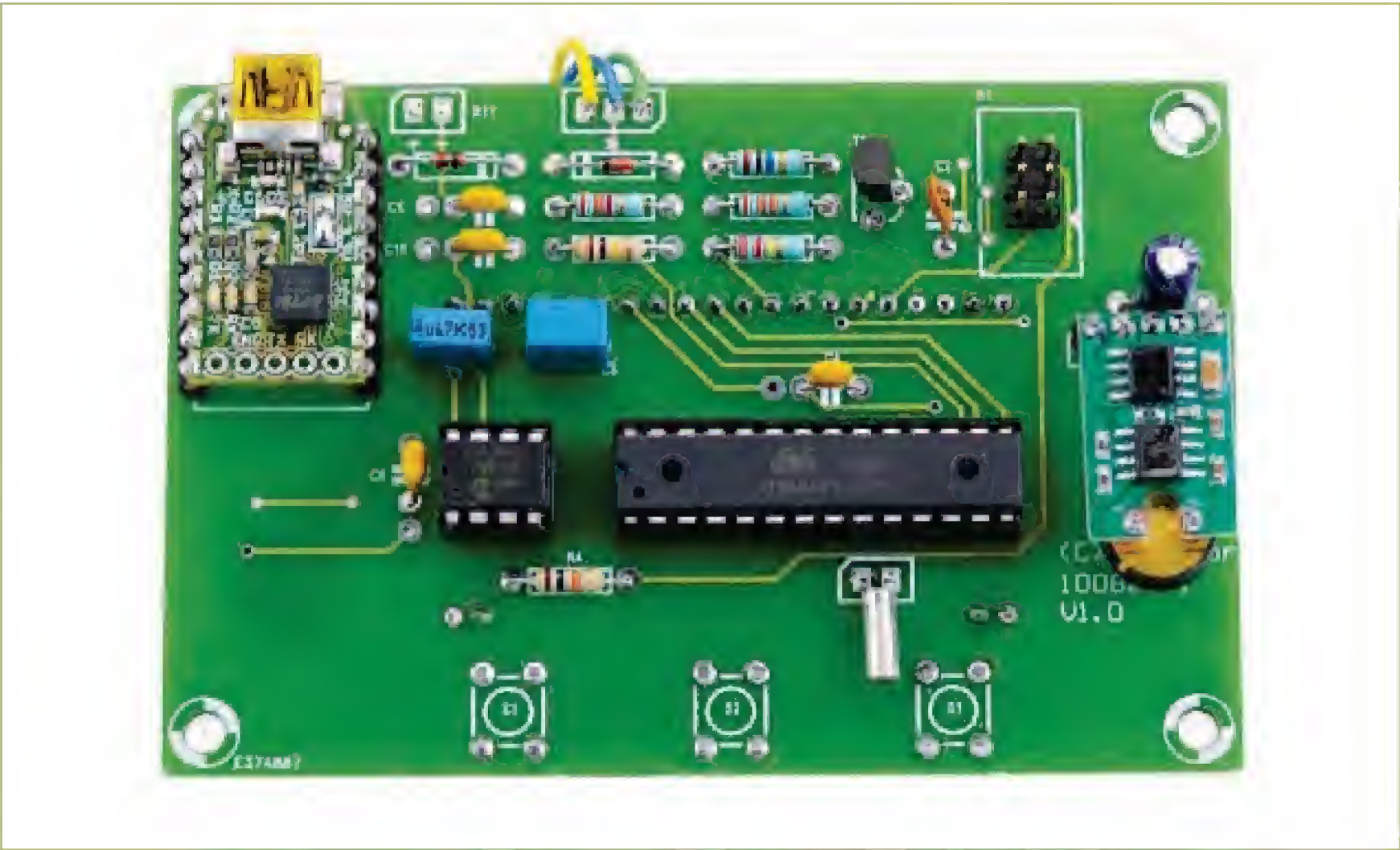
Figure 3. Component side of the Elektor labs prototype board.
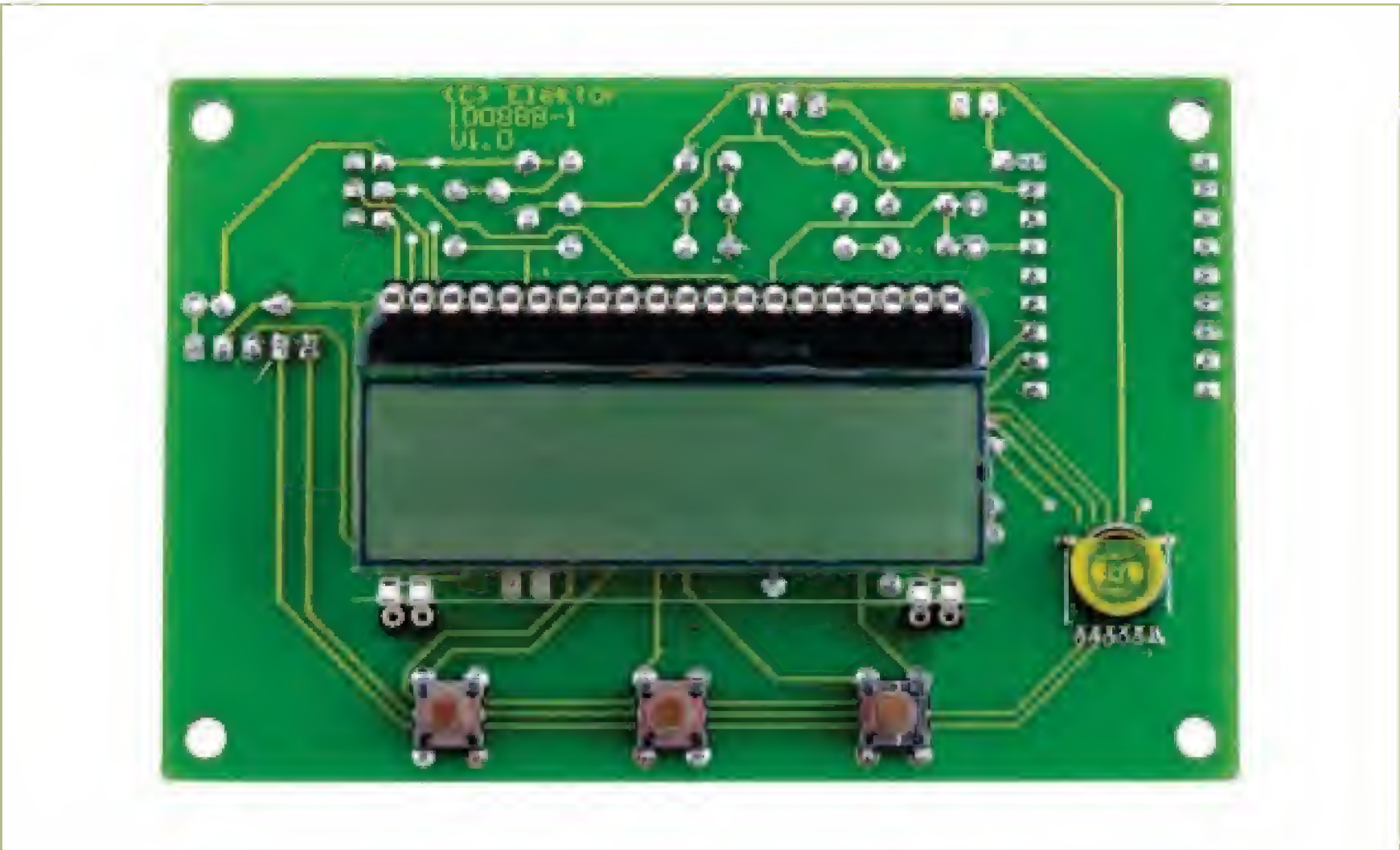
Figure 4. The LCD module and the pressure sensor are mounted on the underside of the board.

Table 1 gives an overview of the functions of each button.

Button S1 cycles between various function modes (numbered 0 to 4 and corresponding to the rows of the table) for S2 and S3. Each press of S1 advances to the next mode. Each row of the table indicates what S2 and S3 do in that mode. **Figure 5** shows the appear-

Table 1. Settings and functions using buttons S1 to S3		
Function mode selected by S1	**Function of S2**	**Function of S3**
0: normal display	display pressure reading	display humidity reading
1: adjust time	increment hours	increment minutes
2: adjust M and N	increment M from 0 to 6	reset N
3: UART control	continue	exit
4: display readings	continue	exit
M = readings per hour; N = count of readings		

Table 2. Interactive commands for data transfer

Command	Action
h =help	print available commands as: # h=help/a=show-p/p#=print#/m#=set-fm#/c=clear/x=exit
a =show-p	print count of readings taken

p# =print#	p0 prints the readings with labels:
	`123 12:30:00 T= 25.6 degC H=43% P= 987.6 hPa`
	p1 prints just the raw numbers:
	`123 12:30:00  256 43  9876`
	p2 prints the time value in hours and the other values without labels, making the output suitable for loading into gnuplot:
	`68.50 256 43  9876`

m# =set-fm#	set number of readings per hour, from 0 to 6 m0 no readings taken m6 six readings per hour (i.e., one every ten minutes)n
c =clear	reset reading count N
x =exit	close serial connection

Table 3. Current drawn by entire system at 3.3 V

CPU frequency	Sleep mode	Current drawn
8 MHz	none	5.8 mA
1 MHz	none	2.4 mA
1 MHz	IDLE	2.0 mA
1 MHz	POWER_SAVE	1.5 mA

Fuse settings for ATmega88

Fuses:	EXT.	0xF9	8 MHz internal oscillator (divided by 8, hence CPU clock is 1 MHz)
	HIGH	0xDF	CKDIV8 enabled, Brown-Out disabled
	LOW	0x62	65 ms startup

ance of the display in 'normal display' mode (mode 0).

To transfer stored readings to a host PC or laptop first connect it to the module using a USB cable. Driver installation is covered in the article on the BOB-FT232R serial-to-USB module elsewhere in this issue.

The microcontroller can detect whether the 3.3 V supply is available from the BOB-FT232R module using port pin PD6.

The PC can retrieve data from the unit over a virtual serial port using the interactive commands described in **Table 2**, for example with the help of HyperTerminal or HTerm.

Power supply

The weather logger operates from a supply voltage of nominally 3.3 V, which, when it is powered over USB, is available at pin 17 (VCC) of the BOB-FT232R serial-to-USB converter module. For stand-alone operation the unit can be powered via BT1 using a supply voltage between 3.5 V and 30 V and an LDO regulator (IC3). Diode D1 protects against accidental reverse polarity connection.

It is important to employ various current-reduction measures to maximise battery life in stand-alone operation. The ATmega88 microcontroller is responsible for the majority of the power consumption, and so we concentrate our efforts there. The measured current draw of the entire circuit at 3.3 V under various conditions is shown in **Table 3**. When using a battery pack consisting of three alkaline or NiMH AA cells (i.e. 4.5 V or 3.6 V; nominal capacity of between 2 Ah and 3 Ah) the weather logger can
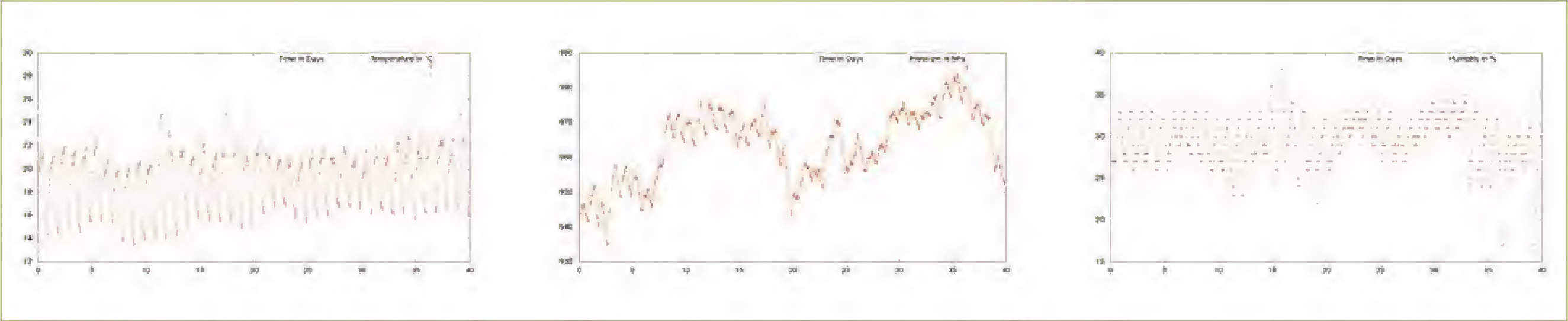
Figure 5. Changes in temperature, humidity and atmospheric pressure can be shown graphically using a program such as gnuplot.

record data for at least 1000 hours. Instead of three AA cells, feel free to employ a single Li-Ion or LiPo cell supplying 3.6 V or 3.7 V respectively.

One power-saving measure is to carry out only one measurement per minute rather than one every second. At one-minute intervals we update only the time, the record count and the elapsed hours timer on the display. The device goes into this mode when no button is pressed for a period of 100 seconds.

Measurements show that the system draws an extra current of about 2.5 mA for about 120 ms when taking a reading; in between these peaks are peaks lasting just 10 ms corresponding to the 1 Hz interrupt.

The device is switched from POWER_SAVE mode into IDLE mode for the fifty-ninth second of the minute. This enables the I/O clock in the microcontroller, which is needed in order to be able to count the pulses from the humidity sensor: the final reading is then available in the sixtieth second. This power-saving trick is only effective if the ATmega88 is operated with a 1 MHz CPU clock: the necessary fuse settings are given in the text box.

Displaying the results

The free software package gnuplot can be used to help display the results on a PC in graphical form. **Figure 6** shows examples of image files that can be created using this software, including temperature, humidity and pressure graphs.

The steps required to produce such graphs, including the transfer of data to the PC using HyperTerminal or HTerm, are given below. The filenames used correspond to the example.

1. Set up HTerm on the PC to read data over USB as follows:
COM3 9600 baud 8N1
newline at: CR
send on enter: CR
storage format: RAW

2. Put the weather logger into USB mode: connect the weather logger to the computer with a USB cable;
press the 'function' button S1 until the message 'UART-control>USB' appears;

press S2 for 'continue': the message 'USB-UART active' should appear.

3. In HTerm give the command 'p2'.
Data transfer will start with the format:
Hours Temperature*10 Humidity Pressure*10
Stop data transfer by typing 'x'.
Save the received data as 'pltdata0.txt'.

4. Plot the data using the Windows version of gnuplot ('wgnuplot').
Open the file 'plotfile0.plt', which contains the plotting commands for wgnuplot that will result in the creation of the following three image files:
ptemp0.png (temperature graph)
ppres0.png (pressure graph)
phumid0.png (humidity graph)

These three image files can be opened in a graphics editor or included in a word processor document using Word or OpenOffice. An illustrated manual is contained in the archive file for this project [8]

(100888)

Pressure readings

The weather logger measures and stores instantaneous values of atmospheric pressure. The values depend not only on the weather but also on altitude and other factors such as temperature and humidity. The pressure values published by weather services are corrected to give the effective atmospheric pressure at sea level (0 m altitude), based on the 'International Standard Atmosphere' values for temperature, humidity and so on. For example, the 'base atmospheric pressure' at sea level is defined as 1013.25 hPa and the 'base temperature' as +15 °C. The pressure values reported by the logger can be converted approximately to sea-level values using the barometric formula

$$p(0) = p(h) / (1 - 22.558 \times 10^{-6} h)^{5.255}$$

where h is the altitude in metres, $p(h)$ is the measured pressure at altitude h measured in hectopascals, and $p(0)$ is the corrected pressure at sea level.

There is a simpler approach: to a good approximation at low altitudes (up to say 2000 m) pressure falls off linearly at about 1 hPa for every 8 m of altitude. So, for example, a pressure of 970 hPa recorded at an altitude of 400 m corresponds to an equivalent sea-level pressure of about 1020 hPa.

Internet Links

[1] www.elektor.com/110376

[2] www.atmel.com/dyn/resources/ prod_documents/doc2545.pdf

[3] www.lcd-module.com/eng/pdf/doma/ dog-me.pdf

[4] www.elektor.com/110533

[5] www.atmel.com/dyn/resources/ prod_documents/doc1981.pdf

[6] www.elektor.com/071125

[7] ww1.microchip.com/downloads/en/ devicedoc/21754h.pdf

[8] www.elektor.com/100888

About the author

Wilfried Wätzig's first encounter with programming was using ALGOL on the (then current) Zuse Z22 and Electrologica X1 machines during his physics studies. Later he developed interface hardware and data processing software for process control computers in a university environment. Finally he worked as a system administrator at the computing service of a technical college and now, in retirement, maintains a keen interest in recent developments in hardware, software and information technology.

I²C Sensors For temperature, atmospheric pressure and humidity

By Ernst Krempelsauer (Elektor Germany Editorial)

Most sensors are inherently analogue and in general it is necessary to amplify, compensate and calibrate their outputs before digitising them. The sensor modules described here have everything built in: a microcontroller connected to the module can fetch the calibration data needed to correct the digitised results over the I²C bus, meaning that no calibration is necessary.

The signal conditioning circuitry built into sensors with a digital output saves development effort, board space and cost. Also, because the sensor element and the conditioning circuitry are closely linked both electrically and thermally, performance of the sensor improves too. The two sensor devices from Hope Microelectronics [1] described here are a frugal choice from the point of view of both cost and power consumption, and both are available from the Elektor shop [2].

The HP03S pressure and temperature module

The HP03S module contains a piezoresistive pressure sensor which can be viewed electrically as four resistive elements in a bridge configuration. As the block diagram in **Figure 1** shows, this conventional pressure sensor is connected directly to a signal conditioning circuit comprising an input multiplexer, an A/D converter and an I²C interface. Also provided on the module a 24C02-compatible EEPROM, which contains calibration data for the sensor pre-loaded by the manufacturer.

Pressure is determined by measuring the voltage across the bridge, which is digitised by the ADC; for temperature measurement the (temperature-dependent) total top-to-bottom resistance of the sensor bridge is measured by making it appear as one arm of a new bridge circuit that includes resistors R1 to R4.

The A/D converter is a 16-bit sigma-delta type with an overall effective resolution of 14 bits. Further technical details can be found in the 'Features' text box.

The datasheet claims the following absolute atmospheric pressure measurement accuracy over the normally-encountered range from 750 hPa to 1100 hPa:

HP03S pressure and temperature module features

- Pressure measurement range: 300 hPa to 1100 hPa
- Temperature measurement range: −40 °C to +85 °C
- Supply voltage: 2.2 V to 3.6 V (3 V typical)
- Current consumption: 1 µA (standby), 500 µA (during measurement)
- Operating temperature: −40 °C to +85 °C
- I²C interface SCL maximum frequency: 500 kHz
- MCLK frequency: 30 kHz to 35 kHz (32.768 kHz typical)
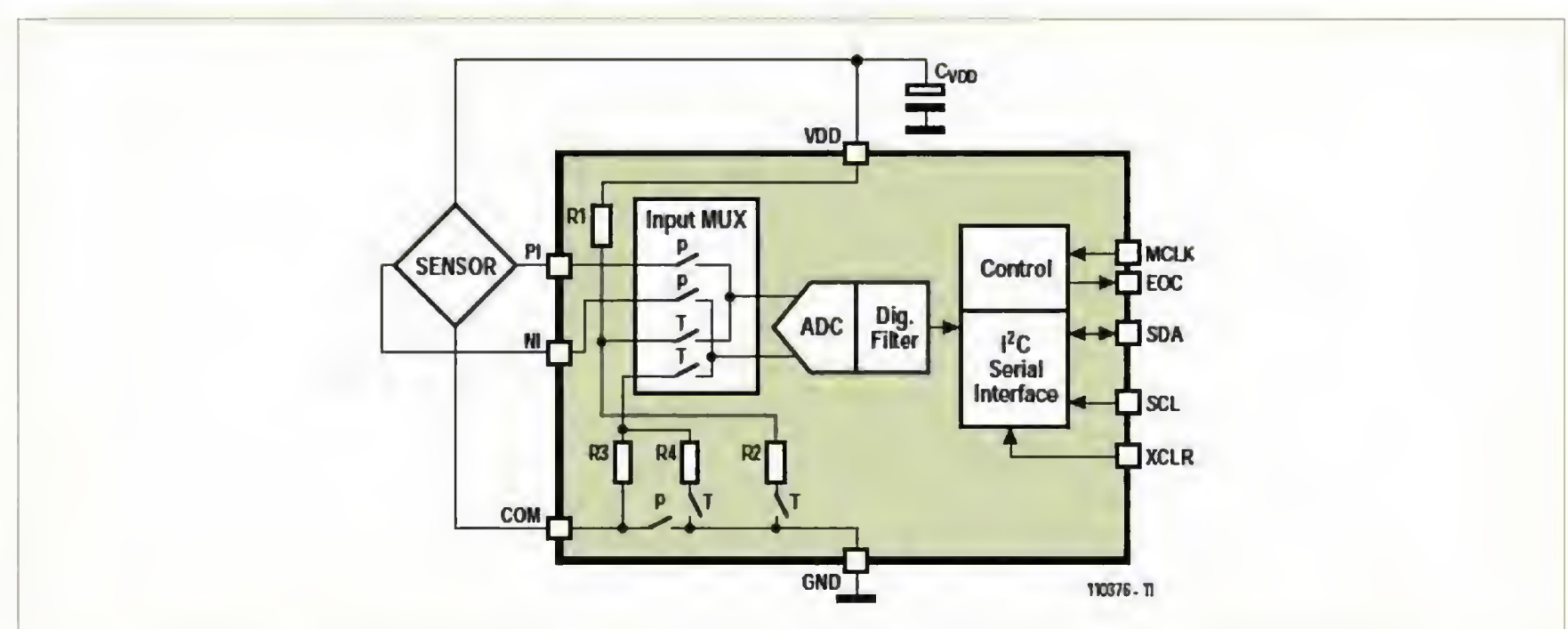
Figure 1. Block diagram of the HP03S pressure and temperature module.

Table 1. HP03S pinout

Signal name	Pin	Function
SCL	1	I²C clock input
SDA	2	I²C data input
XCLR	3	ADC reset input
MCLK	4	Master clock input
VDD	5	Supply voltage +U_B (V_{CC})
VSS	6	Ground (GND)
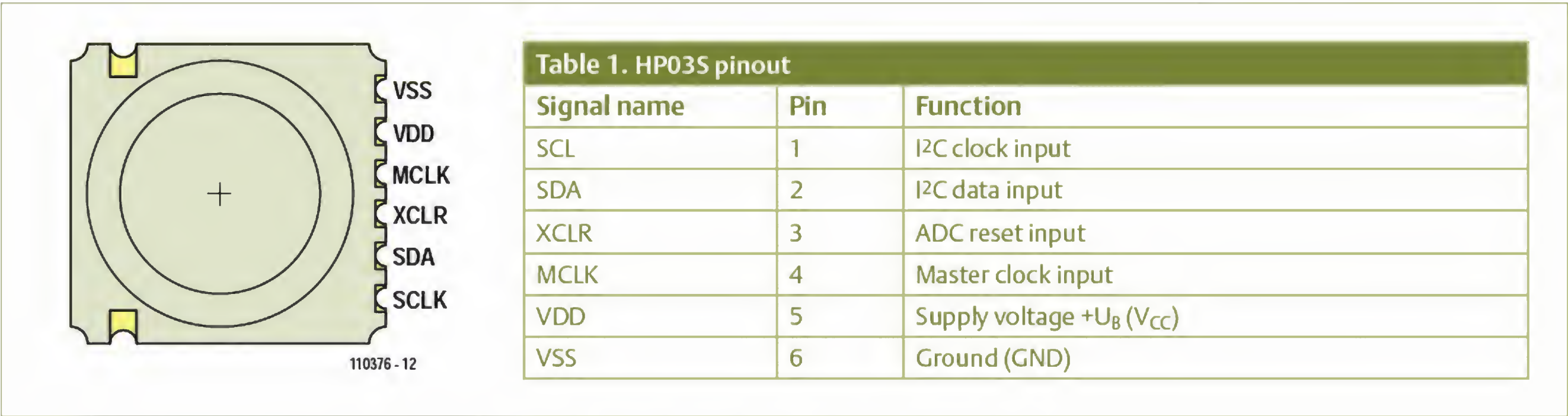
Figure 2. Pinout of the HP03S in its SMD package.

- ±1.5 hPa (HP03SA for temperatures from 0 °C to +50 °C)
- ±3.0 hPa (HP03SA for temperatures from −20 °C to +60 °C)
- ±3.0 hPa (HP03SB for temperatures from 0 °C to +50 °C)
- ±5.0 hPa (HP03SB for temperatures from −20 °C to +60 °C)

Both sensor variants (A and B) offer:
Long-term stability (12 months): 2 hPa typical
Voltage dependence (2.4 V to 3.6 V): ±1.5 hPa
Temperature measurement accuracy (0 °C to +50 °C): ±1.0 °C
Temperature measurement accuracy (−20 °C to +60 °C): ±2.0 °C

The pinout of the device is shown in **Figure 2** and the signals are explained in **Table 1**.
A little care is required with the XCLR connection, which resets the ADC. XCLR should only be taken high to carry out an A/D conversion and when reading out pressure and temperature data; otherwise it should be held low, both in the quiescent state and when reading from the module's EEPROM. The quality of the 32 kHz clock signal MCLK affects the current consumption of the module. The recommendations regarding signal level (2.2 V minimum), edge slew rate and duty cycle (40 % to 60 %) should be adhered to.
The ADC delivers uncorrected readings over the I²C bus: these readings are called 'D1' (nominally pressure) and 'D2' (nominally temperature). Since the pressure readings are highly sensitive to temperature, compensation is required. Seven coefficients and four sensor parameters are stored in the module's serial EEPROM by the manufacturer during calibration. These are labelled as follows.

C1	sensitivity coefficient
C2	offset coefficient
C3	temperature coefficient of sensitivity
C4	temperature coefficient of offset
C5	reference temperature
C6	temperature coefficient of temperature
C7	offset fine tuning
A, B, C and D	sensor parameters

The following computations have to be performed on the raw D1 and D2 values to obtain values for temperature (T), offset (OFF), sensitivity (SENS) and pressure (P). The arithmetic can easily be done in a microprocessor and no floating-point operations are needed.

$$dUT = D2 - C5$$

$$T = 250 + dUT \times C6 / 2^{16}$$
temperature in Celsius ×10

$$OFF = (C2 + (C4 - 2048) \times dUT / 2^{14}) \times 4$$

$$SENS = C1 + C3 \times dUT / 2^{10}$$

$$X = SENS \times (D1 - 7168) / 2^{14} - OFF$$

$$P = X \times 10 / 2^5 + C7$$
pressure in hPa ×10

A practical example application for this sensor, connected to the I²C interface of a microcontroller, is the long-term weather logger circuit presented elsewhere in this issue of *Elektor* [3]. The source code for that unit's firmware, written in C, is available at [3] for free download. From the comments in the code it is easy to see where the coefficients and parameters are read out from the module's EEPROM, where the ADC is read, and where the compensation

calculations are carried out. Alternatively, the manufacturer has a programming guide with an application example [4] written in C for an 8051 microcontroller.
The EEPROM and the ADC have different addresses on the I²C bus: 0xA1 to read the EEPROM and 0xEE to read the ADC. The datasheet of the 24C02 [5] gives a good introduction to bus timing. Remember that it essential to take XCLR high when reading the ADC.
The module outputs all data in binary format. The first word read after power-up should be discarded: words from the second onwards can be used safely.
In the HP03S datasheet [6] there is a detailed example showing how to calculate pressure and temperature using the sensor's coefficient values and parameters. When using the sensor as part of an altimeter it is desirable to have a higher pressure resolution, calculating the pressure as

$$P = X \times 100 / 2^5 + C7 \times 10.$$

It is also clearly explained in the datasheet that the sensor is not designed for use in safety-critical applications, especially those where a failure of the sensor could put human health or life at risk.

HH10D humidity sensor

The Hope Microelectronics HH10D humidity sensor module uses a capacitive transducer, shown in the circuit diagram (**Figure 3**) as a variable capacitor, which determines the frequency of an oscillator built around an ICM7555 CMOS timer. The output signal from this 'capacitance-to-frequency converter' appears on the FOUT pin of the module (see pinout in **Figure 4**). The I²C interface of the HH10D is used only to read its 24C02 EEPROM, which again is used to store calibration values for the sensor. Each sensor is individually calibrated by

HH10D humidity sensor module features

- Relative humidity measurement range: 1 % to 99 %
- Accuracy: ±3 %
- Resolution: 0.3 % to 0.05 % (0.08 % typical)
- Reproducibility: ±0.3 %
- Reaction time: 8 seconds
- Hysteresis: 1 %
- Long-term stability: ±0.5 %
- Supply voltage: 2.7 V to 3.3 V (3 V typical)
- Current consumption: 120 µA to 180 µA (150 µA typical)
- Operating temperature: –10 °C to +60 °C
- Output frequency (FOUT): 5 kHz to 10 kHz (6.5 kHz typical)
- I²C interface: M24C02BN compatible
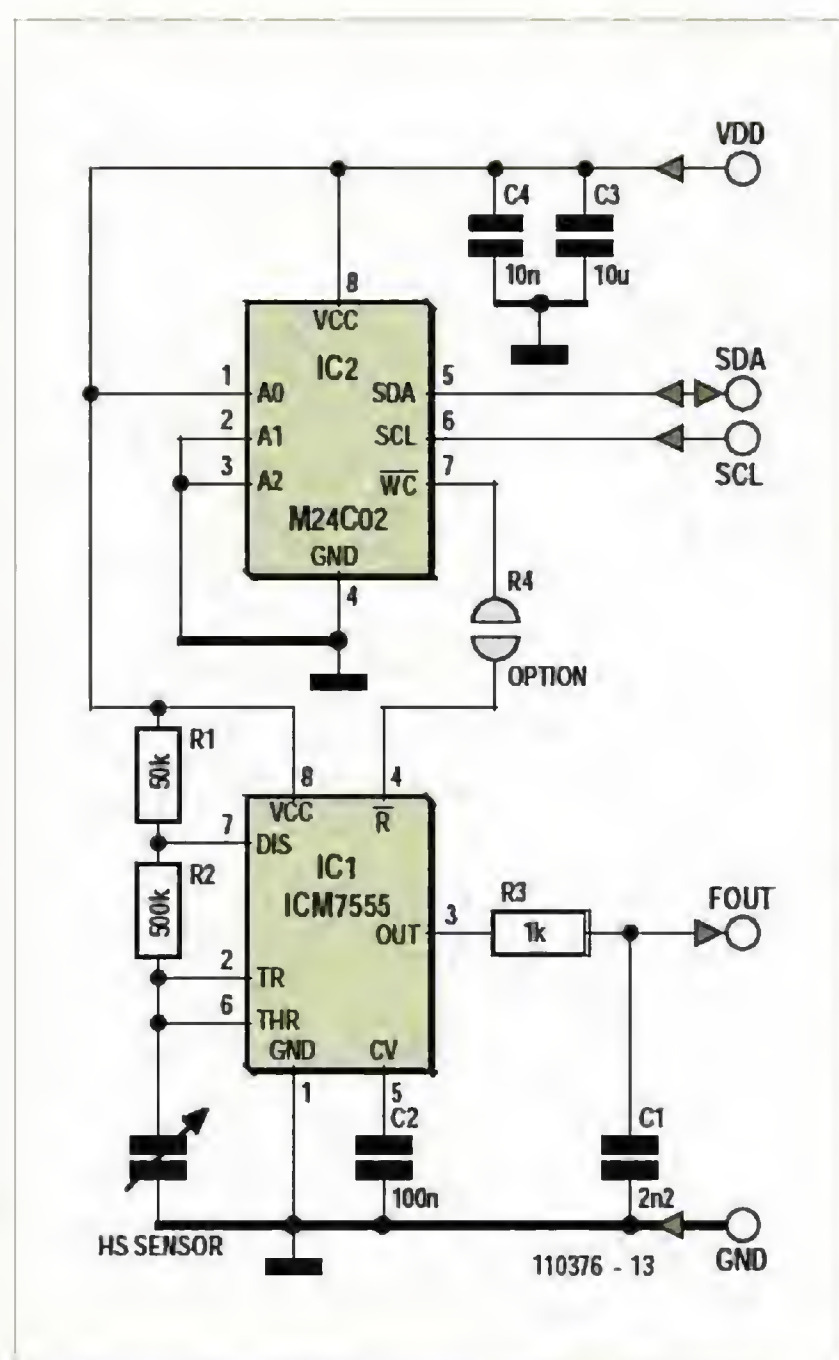
Figure 3. Circuit diagram of the HH10D humidity sensor module.

Figure 4. Pinout of the HH10D, available as a plug-in daughter module with a pin header.

the manufacturer in two different humidity-controlled environments.

Using a capacitive sensor has the advantage that it reacts quickly to changes in ambient humidity. The sensor module also offers a particularly high resolution and, thanks to the stored calibration values, very high precision despite its low power consumption. The main features and technical specifications are given in the text box on the HH10D.

To compute a humidity value, a microcontroller connected to the SCL, SDA and FOUT

pins must first read out the two calibration coefficients stored in the serial EEPROM (two bytes each). These are as follows.

SENS sensitivity coefficient
(EEPROM address 10)

OFF frequency offset
(EEPROM address 12)

The address of the EEPROM itself is fixed at 01.

The frequency FREQ of the oscillator can be measured using a timer/counter in the microcontroller. This value can be converted into a relative humidity (RH), expressed as a percentage, as follows:

$$RH = (OFF - FREQ) \times SENS / 2^{12}$$

Again, a practical example showing how to use this sensor can be found in the source code accompanying the weather logger article [3]. The datasheet for the humidity sensor can be downloaded at [7].

Other I²C sensors

A wide range of other sensors with I²C interfaces is available. As the examples described above show, the implementations vary considerably from one sensor type to another and there is no alternative to studying the datasheet in each case.

Hope Microelectronics also produces the HDPM01 sensor module, an interesting device that combines an HM03 pressure sensor with a two-axis compass [8], again with an I²C interface. A different I²C compass sensor is used in the Elektor ATM18 project [9] elsewhere in this edition. Sensirion [10] is also worth mentioning for their I²C temperature and humidity sensors. For temperature sensors in particular there is a wide choice of I²C bus devices, for example from NXP [11] and Maxim [12].

Other well-known temperature sensors are the National Semiconductor LM76, the Maxim (formerly Dallas) DS1621 and DS1631, and the Texas Instruments TMP100.

(110376)

Internet Links

[1] www.hoperf.com

[2] www.elektor.com/110376

[3] www.elektor.com/100888

[4] www.hoperf.com/upload/sensor/
HP03_code.pdf

[5] http://ww1.microchip.com/down-
loads/en/DeviceDoc/21202J.pdf

[6] www.hoperf.com/upload/sensor/
HP03S.pdf

[7] www.hoperf.com/upload/sensor/
HH10D.pdf

[8] www.hoperf.com/sensor/app/
HDPM01.htm

[9] www.elektor.com/110389

[10] www.sensirion.com/en/01_humidity_
sensors/00_humidity_sensors.htm

[11] http://ics.nxp.com/products/
i2cthermal

[12] www.maxim-ic.com/products/
thermal-management

Milkymist SoC
An open-source programmable chip

You will certainly be familiar with 'systems on chip' (SoC). These are 'big microcontrollers' that incorporate a powerful microprocessor, an SDRAM controller, and various peripherals depending on the applications being targeted — sometimes even graphics processing accelerators for OpenGL ES.
To the curious electronics enthusiast, these circuits are 'black boxes': we don't know much about how they work, and the enormous resources needed to design and manufacture them are more than enough to discourage amateur initiatives to reproduce them.

By Sébastien Bourdeauducq (France)
Founder of the Milkymist project

However, cheap and increasingly dense and powerful FPGAs are now making it possible for any skilled, motivated person to have a go at the upper layers of designing an SoC — the layers involving the computer architecture and the code written in a hardware description language (typically VHDL or Verilog). This goes beyond simple intellectual curiosity, as going about it this way allows you to incorporate specific peripherals onto your chip easily, by taking advantage of the FPGAs' flexibility and computation power. It would even be possible to envisage a large 'open source' community, comparable to the one around Linux, if the major semiconductor manufacturers join in the game (though, without wishing to carp, that prospect still seems a long way off).

This article presents Milkymist SoC, a system-on-chip environment, the source code of which, written in Verilog, is almost entirely under GNU GPL licence, after the fashion of Linux. For the moment, we're not going to go into the details of its design but are going to confine ourselves to programming it — as might be done with any other more conventional platform; the idea is to demonstrate that there is a perfectly feasible alternative to using closed SoCs. Readers interested in the architecture and the internal operation of Milkymist SoC may consult the documentation and the code available on line, or wait for forthcoming articles.

First contact

If you visit the project website [1], you may be surprised to find a video synthesizer intended for VJs (video jockey), clubs, and musicians. This device (**Figure 1**) allows psychedelic, interactive visual effects to be added to accompany a musical performance, using

for example the image of a dancer filmed live by a camera and processed using an array of programmable effects.
This is actually the first application conceived by the project, implementing the *Flickernoise* video synthesis software developed for this platform. Unlike many 'free software' companies, the Milkymist business model is not to bill services associated with free code (facilities management, on-line services, engineering consultancy, etc.), but to develop from A to Z and sell a consumer product employing freely available techniques.
The project goes out of its way to employ as few proprietary components as possible. Thus some techniques initially developed within the context of Milkymist can be found in applications that have nothing to do with graphics or video synthesis. For example, NASA's CoNNeCT experiment, due to be installed aboard the international space station in January 2012, contains a software radio system that re-uses the SDRAM controller developed for Milkymist and made available on free download on the Internet. Again, the firmware debugging system (based on GDB) developed for the Milkymist platform figures in the design stage for use in a control system for the particle accelerators at CERN and GSI.
A beta version of the Milkymist One video synthesizer is currently available as a development kit from specialist retailers, like Hackable Devices [1]. This includes a perfectly viable development board for FPGA or firmware – the beta version in fact refers to the fact that the Flickernoise software still contains certain bugs and certain functions are missing, so it is still not ready for the general public.
The Milkymist One platform is based on a Spartan-6 FPGA from Xilinx (XC6SLX45), around which there are numerous peripherals: 128 MB of DDR SDRAM, 32 MB of NOR flash memory, VGA output

(© 2011 John Lejeune)

Internet Links

[1] www.milkymist.org

[2] www.hackable-devices.com

[3] www.cygwin.com

[4] www.milkymist.org/snapshots/latest/

[5] www.qemu.org

[6] http://lists.milkymist.org

(resolution up to 1,280 × 1,024), 10/100 Ethernet, PAL/SECAM/ NTSC video input, memory board (enables the storage capacity to be readily expanded up to several gigabytes), AC'97 audio, two DMX512 (RS-485) ports, 36 kHz infra-red receiver (for example RC5), two MIDI ports, and two USB host ports. For those who like to tinker, the board is fitted with an expansion port with 12 lines in

Figure 1. The Milkymist One with its case.
(© 2011 Sharism at Work Ltd.)

3.3 V logic. Perhaps not a great deal compared to a typical development board, but this does all the same allow some interesting extensions – all the more so because, correctly programmed, the XC6SLX45 allows input/output frequencies of up to 1 GHz per line. The FPGA contains the whole of the Milkymist SoC (**Figure 2**). This is made up of a LatticeMico32 microprocessor core (32-bit RISC), IP blocks allowing all the Milkymist One peripherals to be controlled from the software, and graphics acceleration. Apart from the LatticeMico32 core, all the rest of the Verilog code has been developed specifically for Milkymist and placed under a GNU GPL licence.
It is possible to port the Milkymist SoC to other FPGA development boards. Whether they're from Altera, Lattice, or Xilinx is not very important; special emphasis has been placed on the portability of the SoC's Verilog code. However, adapting the memory system to another FPGA family or another type of SDRAM requires special technical skills, and many porting attempts have failed because of this tricky point.

And lastly, if you don't have a development board for the moment, you'll be able to carry out the manipulations described in this article by means of the QEMU emulator. This will be explained later on.

Getting started
We'll assume you are the proud owner of a Milkymist One. Connect up the AC power supply, an SVGA screen, and a USB mouse and keyboard. Press the on button (in the middle). After a dozen or so seconds, the Flickernoise appears on the screen (**Figure 3**).
We'd encourage you to explore its functions a bit, just to get an idea of the power of the platform.
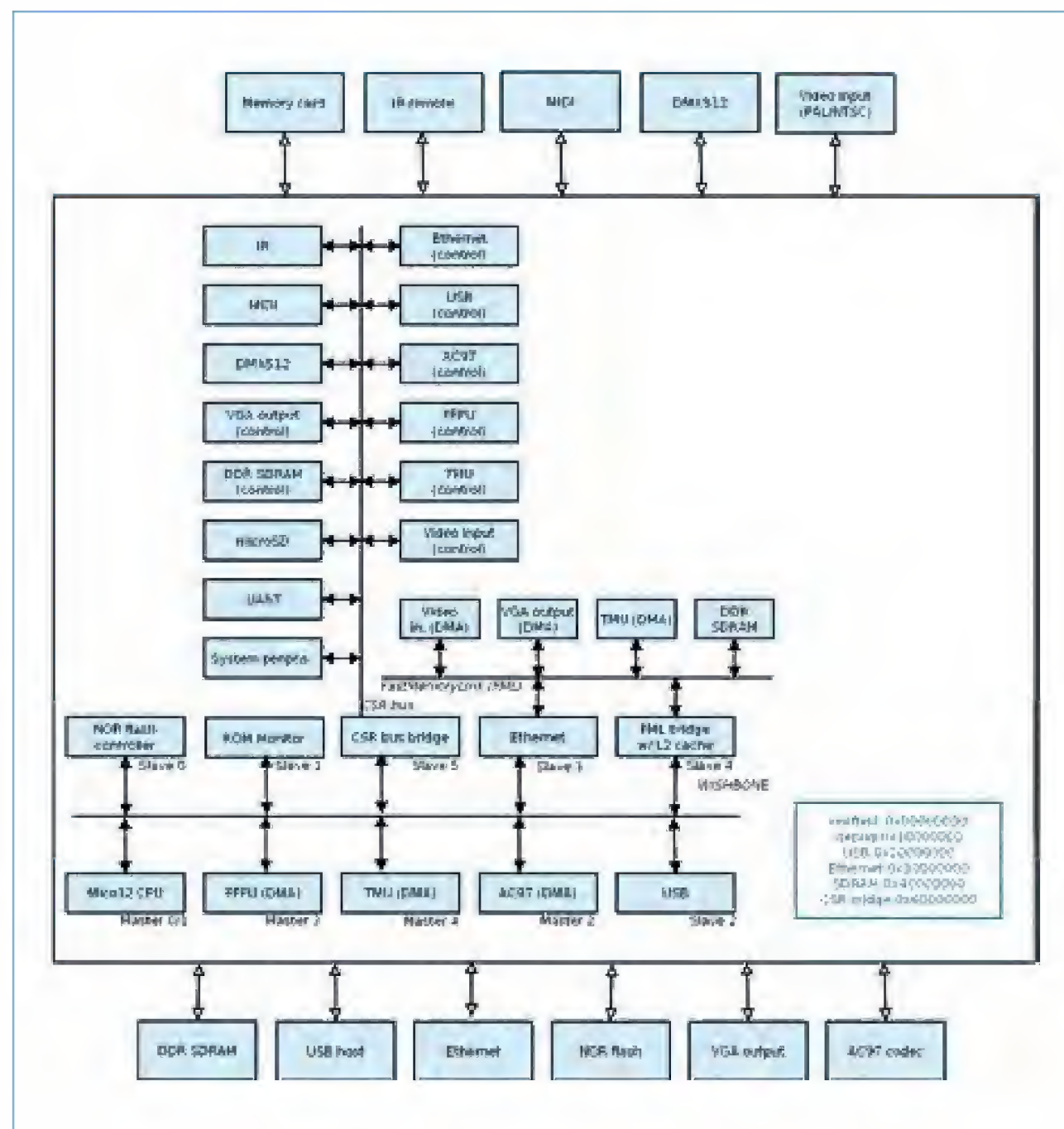
Figure 2. Internal architecture of Milkymist SoC.

When you've finished, click on Shutdown then Reboot and hold down the Esc key as it reboots. Instead of Flickernoise, you should get the rather more Spartan bootloader interface, named BIOS (**Figure 4**).

Type "help" then Enter. The bootloader gives you a list of the commands available (**Figure 5**). From the list of commands, we'll just note the ones that will enable us to run the final programme from the various media:

– **flashboot** runs the software stored in the NOR flash memory. This command is executed by default, and this is the way Flickernoise is run automatically.
– **netboot** downloads the program via TFTP from the Ethernet. Thanks to the speed of Ethernet, this method is particularly useful when debugging binaries that run to several megabytes, like Flickernoise or the Linux core.
– **fsboot** runs the program stored on the memory board.
– **serialboot** downloads the program from a serial connection. This is the method we're going to be using next.

Now we're going to see how to write such a program.

Installing the development tools

The development tools are mainly intended to operate under a Linux system. If you are under Windows, you certainly ought to be able to use them via Cygwin [3]. For Apple users, several people have

contributed a number of tools in MacPorts, but at the time of writing, this is still incomplete.

We're going to concentrate on the RTEMS operating system. The other choices currently available for developing on Milkymist SoC are uClinux (a version of Linux for systems without an MMU) and in bare metal, without an operating system, as for a microcontroller. RTEMS (Real Time Executive for Multi-processor Systems) is an open-source real-time operating system for embedded systems. It has been being developed since 1988 at the initiative of the US Army. The acronym RTEMS originally stood for Real Time Executive for Missile Systems, quickly changed to Real Time Executive for Military Systems, before taking on its current meaning.

RTEMS is designed to be compatible with several standards of API, in particular POSIX. Although it does not offer a memory protection system, RTEMS does offer almost all the POSIX services not connected with this. In POSIX terminology, it might be described as a mono-process, multi-thread system. RTEMS also includes a ported version of the FreeBSD TCP/IP stack and several file systems (MS-DOS, NFS, etc.)

Thanks to this compatibility, it is possible without too much difficulty to implement numerous software libraries from the immense diversity of the world of Linux. This makes it possible to obtain quite a rich a software environment while still keeping a certain lightness compared with embedded Linux. An RTEMS application may easily be less than 150 KB and start in less than a second.

To install the set of tools for development using RTEMS on Milkymist, the simplest thing is to use the binaries for PC under Linux, available from [4] and to be saved into the folder /opt/rtems-4.11. Then update certain environment variables:

$ RTEMS_MAKEFILE_PATH=/opt/rtems-4.11/lm32-rtems4.11/milkymist
$ export RTEMS_MAKEFILE_PATH
$ PATH=/opt/rtems-4.11/bin:$PATH
$ export PATH

You can also easily compile them yourself for your own development machine, thanks to a set of scripts. To do this, first modify your environment as above, and then download the scripts by means of the Git utility:

$ git clone git://github.com/milkymist/scripts.git

Git is a version control system, i.e. a piece of software that lets you properly organize the various modifications made to a code repository and to work efficiently as a team on the same program. This is an excellent quality tool which has been developed by Linus Torvalds to replace the proprietary tool BitKeeper, which had previously been used for the development of the Linux core.

Once the scripts have been downloaded, ensure that you have a folder called /opt/rtems-4.11 (which may be empty) and run them using:

$ make -C compile-lm32-rtems
$ make -C compile-flickernoise milkymist-git-clone
$ make -C compile-flickernoise flickernoise.fbi

This may take several tens of minutes. In actual fact, in addition to the compilation suite based on GCC, a certain number of soft-

Figure 3. Flickernoise screenshot.

ware components will be built to be used and run on Milkymist, in particular:

- the C library and the RTEMS 'core'
- support for the YAFFS2 flash file system
- the encoders and decoders for libpng, libjpeg, openjpeg (JPEG2000), and jbig2dec (JBIG2) images
- the Freetype font rendering library
- the libgd graphics library
- a variant of the liblo OpenSoundControl library
- the MuPDF PDF document rendering system (used for Flickernoise's online help)
- the libcurl multi-protocol network client
- the expat XML parser
- the MTK user interface toolkit

The use of all these libraries would be outside the scope of this article. They are simply mentioned here to give you an idea of the variety of what can currently be implemented on the platform.

Writing and compiling our first program

Now we are armed and ready for the classic "Hello World!" Nothing very new here: open a text editor and simply enter the following code, which you will save with the name hello.c:

```c
#include <stdio.h>
int main()
{
        printf("Hello World!\n");
        while(1);
}
```

However, it's not quite so simple to compile it; here's how it's done using the following command:
$ **lm32-rtems4.11-gcc -O2 -mbarrel-shift-enabled -mmultiply-enabled -mdivide-enabled -msign-extend-enabled -I $RTEMS_MAKEFILE_PATH/lib/include -B $RTEMS_MAKEFILE_PATH/lib -specs bsp_specs -qrtems -o hello hello.c**

If you don't get an error message, the operation has been successful and you ought to have a binary named "hello" in the ELF format. This contains both your "Hello World!" application and the RTEMS core, statically linked. This executable can be run directly on the development board, or in the QEMU emulator.

Testing in QEMU

QEMU [5] is a well-known piece of software that let's you emulate various platforms or to perform virtualization. The latest versions are capable of directly emulating the Milkymist SoC.
So once QEMU is installed, all you have to do is enter the following command to test your binary:
$ **qemu-system-lm32 -M milkymist -nographic -kernel hello**
This should display the famous "Hello World!" Now let's try out the same program on the development board.

Figure 4. The bootloader, called BIOS.
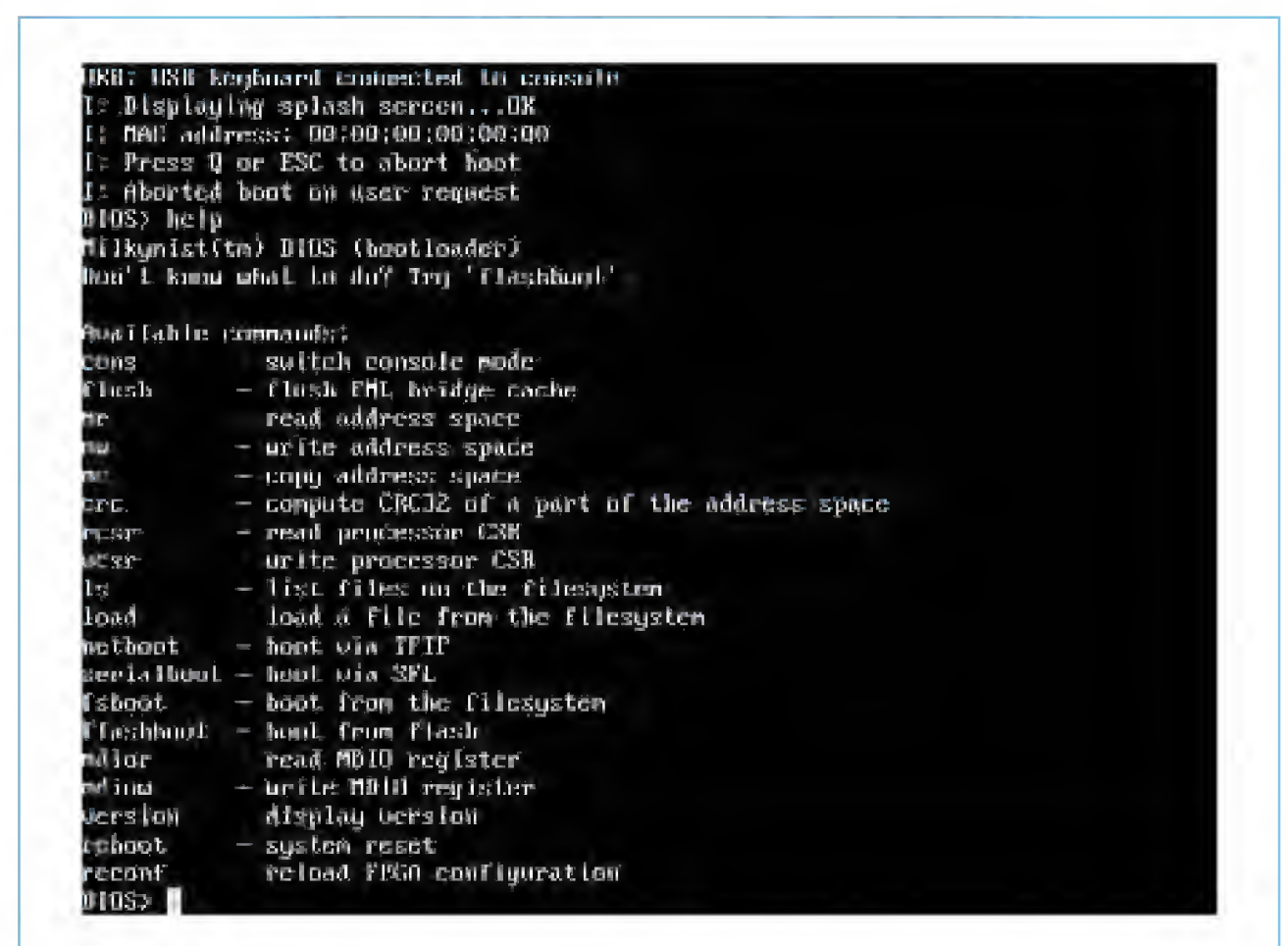
Figure 5. List of commands available.
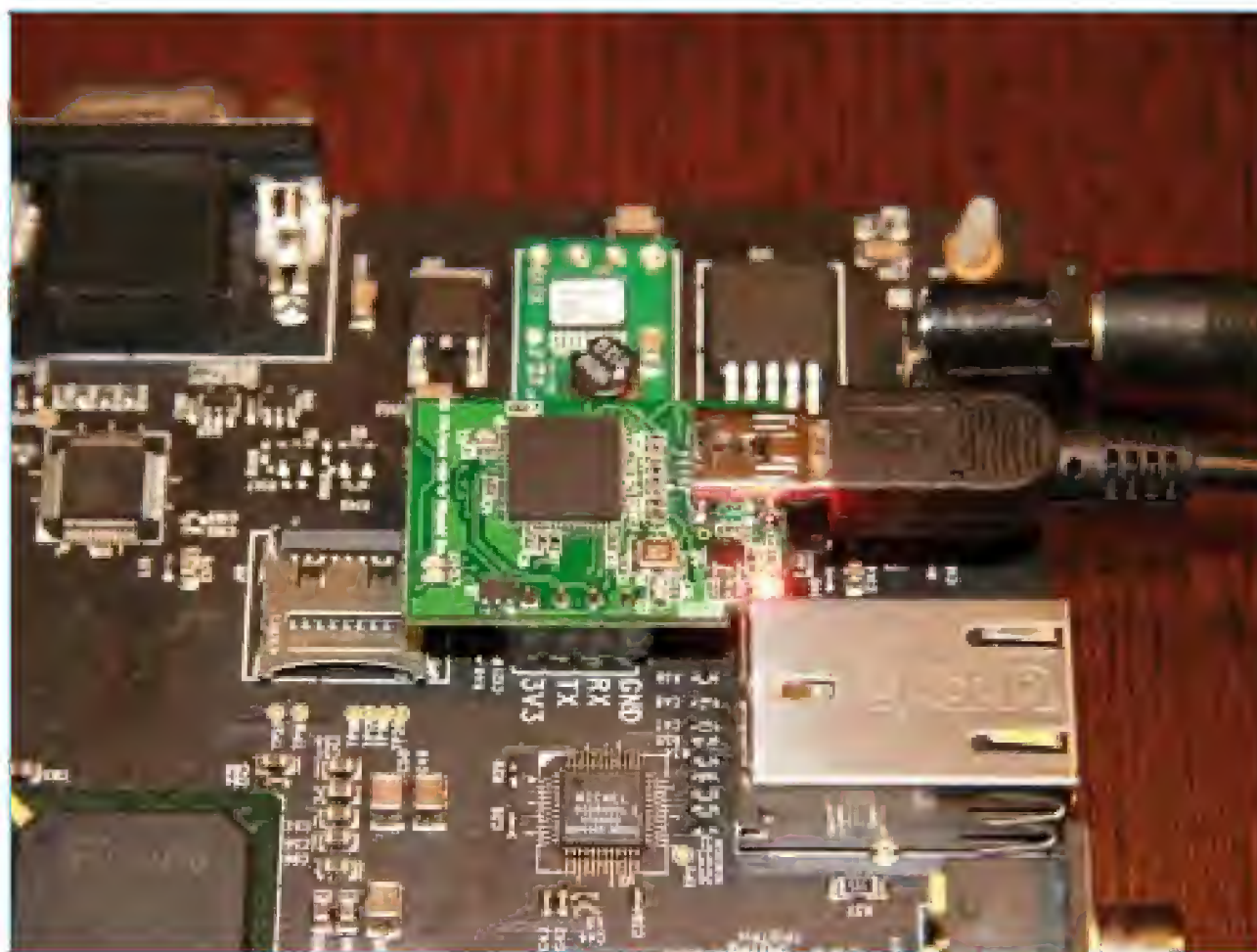
Figure 6. The JTAG + serial adaptor installed on the Milkymist One.
(© 2010 Sharism at Work Ltd.)
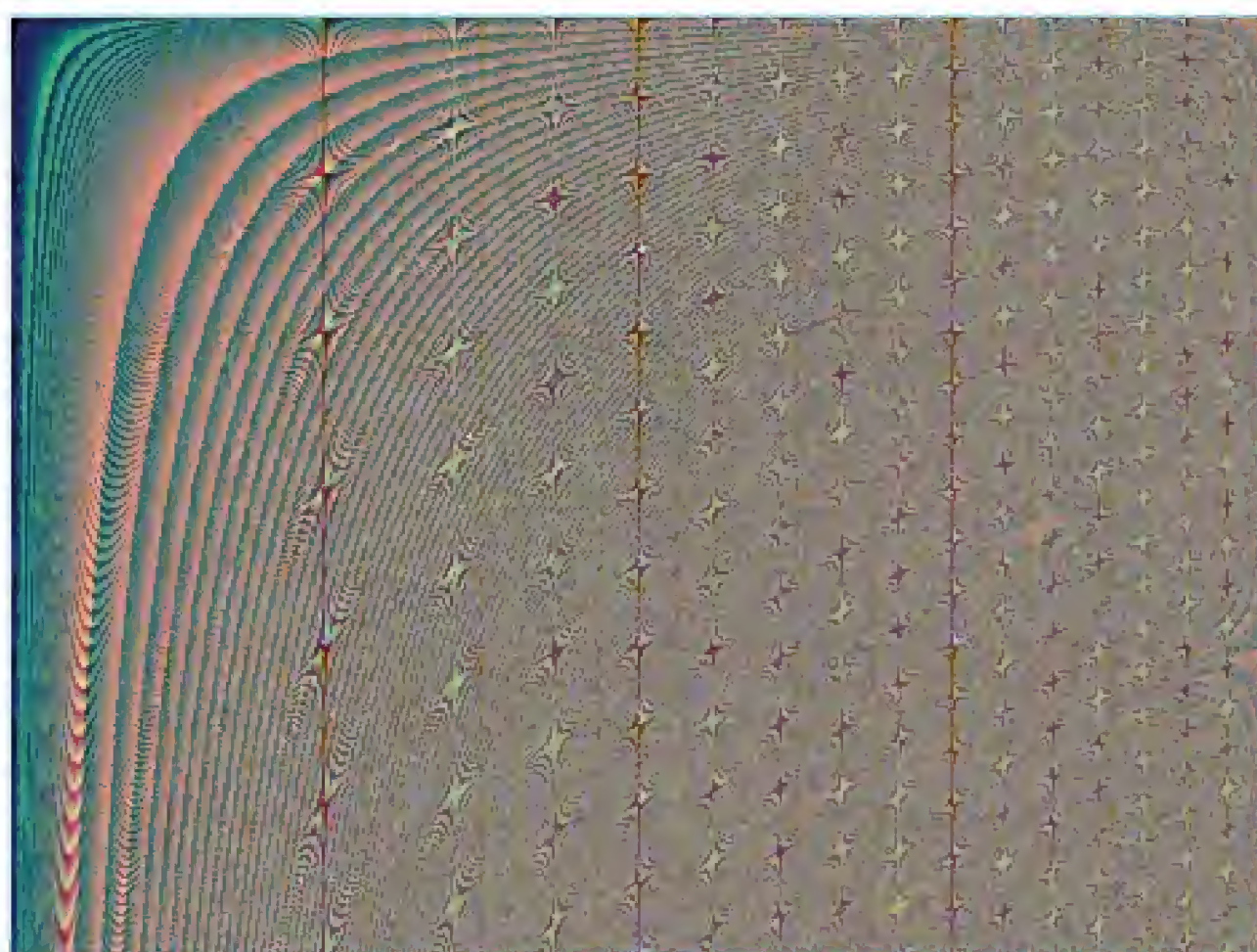
Figure 7. The pattern obtained after numerous iterations of the
command pixel[i] = x * y * x >> 5.

Testing on the development board

We're going to use the serial port to upload our application. It will also serve as a console for displaying the messages sent to printf(). The board is fitted with a 3.3 V serial port, located between the Ethernet and VGA connectors. The pin marked RX is the one on which the board receives the data, and the one marked TX is used by the board for transmission. The GND pin is obviously the ground, and 3V3 is a 3.3 V power rail.

You can use any serial adaptor you choose, as long as it uses 3.3 V levels (not 5 V or RS-232) or the combined serial + JTAG unit (**Figure 6**)

sold with the Milkymist One development kits. This little board plugs onto the Milkymist One's two serial and JTAG connectors and has a USB port for the connection to the PC. Using a recent Linux core, the serial port ought to come up immediately as /dev/ttyUSB0.

For uploading the binary, you'll have to use a utility called *flterm*. This is available in certain Linux distributions, like Fedora. Otherwise, download and compile it manually:

$ wget https://github.com/milkymist/milkymist/raw/master/ tools/flterm.c
$ gcc -O2 -o flterm flterm.c

In order to load your binary onto the board, you must first convert it from the ELF format to a raw binary form. Use the following command for this:

$ lm32-rtems4.11-objcopy -Obinary hello hello.bin

Now run flterm like this:

$ flterm --port /dev/ttyUSB0 --kernel hello.bin

Get the "BIOS" prompt on the board as seen before, and enter the command **serialboot**. Note that you can use the USB keyboard and SVGA screen at the same time as flterm's serial console for dialoguing with the BIOS.
You should obtain the following messages:

```
BIOS> serialboot
[FLTERM] Received firmware download request from the
device.
[FLTERM] Uploading kernel (83476 bytes)...
[FLTERM] Upload complete (9.5KB/s).
[FLTERM] Booting the device.
[FLTERM] Done.
Hello World !
```

Well done, your development environment works! To reboot the development board, all you have to do is press the three buttons together and then release SW3 first.

To take things further...

This article has only skimmed the surface of what it is possible to do. They are plenty of other fields: use of the existing graphics accelerators, video digitizing, acceleration of other computations using the FPGA, development of special I/O interfaces, other programming languages (Lua, Ruby), embedded Linux, in-situ debugging using GDB, and so on.

Send me your comments and suggestions to sebastien@milkymist. org. It will be better to submit questions of a technical nature to the project distribution list [6] so that other people can answer, and the solutions to problems will be archived. The project also has an IRC channel named #milkymist on the Freenode network.

(110447)

Now for a more challenging example: using the video output

Now that we've validated our development system, we're all set to write a slightly more complicated program. Why not a bit of graphics programming, using the SVGA output?

To achieve this, RTEMS provides an interface close to the Linux framebuffer, i.e. it creates a file in /dev upon which the POSIX operations (open, read, write, "ioctl") are possible. The effect of certain of these operations is identical to that under Linux, which can make application porting easier.

The first problem is to enable the video driver. In our first example "Hello World!", we didn't specify an RTEMS configuration, and so the default configuration was used, which does not include the video driver. RTEMS is configured using a series of #define and by including <rtems/confdefs.h>. To add the video driver, all we have to do is define CONFIGURE_APPLICATION_NEEDS_FRAME_BUFFER_DRIVER. Unfortunately, if we use our own configuration in place of the default one, we also have to specify the configuration for the RTEMS's other functions; this is why the end of the program is quite long.

Next we can open the file /dev/fb in our application. The first thing to be done is to define the video mode to be used. This is done using an ioctl call. We're going to choose 1,024 × 768 at 16 bits per pixel. The colour mode is RGB565, i.e. the first five bits (MSB) are for red, the next six for green, and the last five (LSB) for blue.

And lastly, we obtain the framebuffer memory address, via another ioctl call. All we then have to do to display pixels is write into this memory area. The first 1,024 16-bit words correspond to the first line displayed (at the top of the screen). The next 1,024 correspond to the second line, and so on. Generally, a pixel with co-ordinates (x,y) is found at the memory address 1,024 * y + x in the memory area.

All this gives us the following program:

```c
#include <rtems.h>
#include <bsp.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <rtems/fb.h>

rtems_task Init(rtems_task_argument argument)
{
        int fd;
        struct fb_fix_screeninfo fb_fix;
        unsigned short *pixels;
        int x, y;
        int offset;

        fd = open("/dev/fb", O_RDWR);
        ioctl(fd, FBIOSETVIDEOMODE, 2);
        ioctl(fd, FBIOGET_FSCREENINFO, &fb_fix);
        pixels = (unsigned short *)fb_fix.smem_start;
        offset = 0;
        for(y=0;y<768;y++)
                for(x=0;x<1024;x++)
                        pixels[offset++] = x*y*x >> 5;
        while(1);
}

#define CONFIGURE_APPLICATION_NEEDS_CLOCK_DRIVER
#define CONFIGURE_APPLICATION_NEEDS_CONSOLE_DRIVER
#define CONFIGURE_APPLICATION_NEEDS_FRAME_BUFFER_DRIVER
#define CONFIGURE_MAXIMUM_DRIVERS 4
#define CONFIGURE_USE_IMFS_AS_BASE_FILESYSTEM
#define CONFIGURE_EXECUTIVE_RAM_SIZE (16*1024*1024)
#define CONFIGURE_LIBIO_MAXIMUM_FILE_DESCRIPTORS 4
#define CONFIGURE_MAXIMUM_TASKS 2
#define CONFIGURE_TICKS_PER_TIMESLICE 3
#define CONFIGURE_MICROSECONDS_PER_TICK 10000
#define CONFIGURE_RTEMS_INIT_TASKS_TABLE
#define CONFIGURE_INIT_TASK_STACK_SIZE (8*1024)
#define CONFIGURE_INIT_TASK_PRIORITY 100
#define CONFIGURE_INIT_TASK_ATTRIBUTES 0
#define CONFIGURE_INIT_TASK_INITIAL_MODES \
        (RTEMS_PREEMPT | RTEMS_NO_TIMESLICE | RTEMS_NO_ASR | \
        RTEMS_INTERRUPT_LEVEL(0))
#define CONFIGURE_INIT
#include <rtems/confdefs.h>
```

Compile it and test it as we have seen above. If you're using QEMU, remove the option "-nographic".

The value x * y * x >> 5 assigned to each pixel gives the pattern shown in **Figure 7**.

ATM18 Compass
"You'll never walk alone"

By Grégory Ester (France)

Now you can forget all about magnetised needles on their pivots for finding magnetic North. And it doesn't matter if you live in the Southern or Northern hemisphere – all that counts here is that you have both feet firmly on the ground and this little device in your hand.
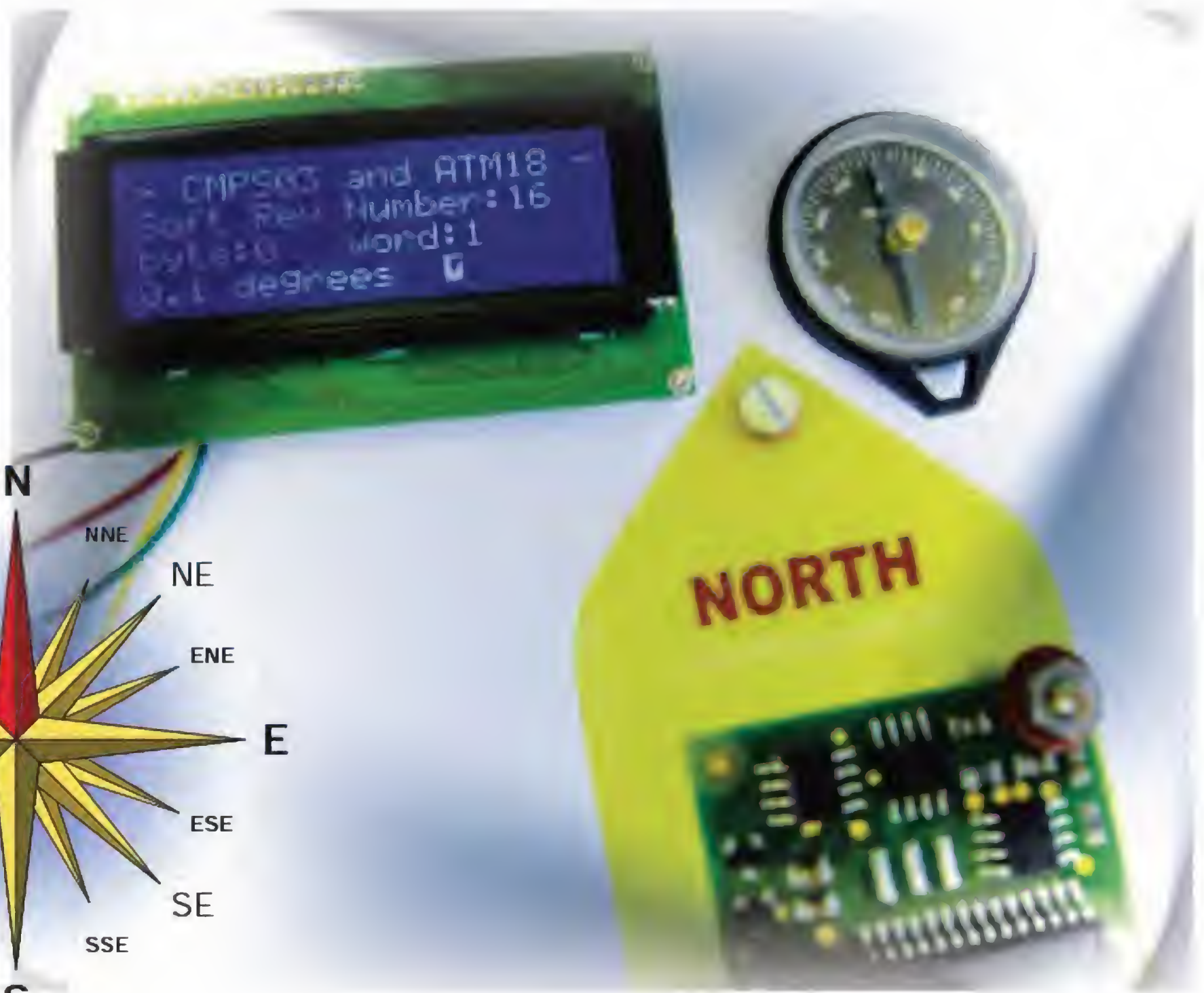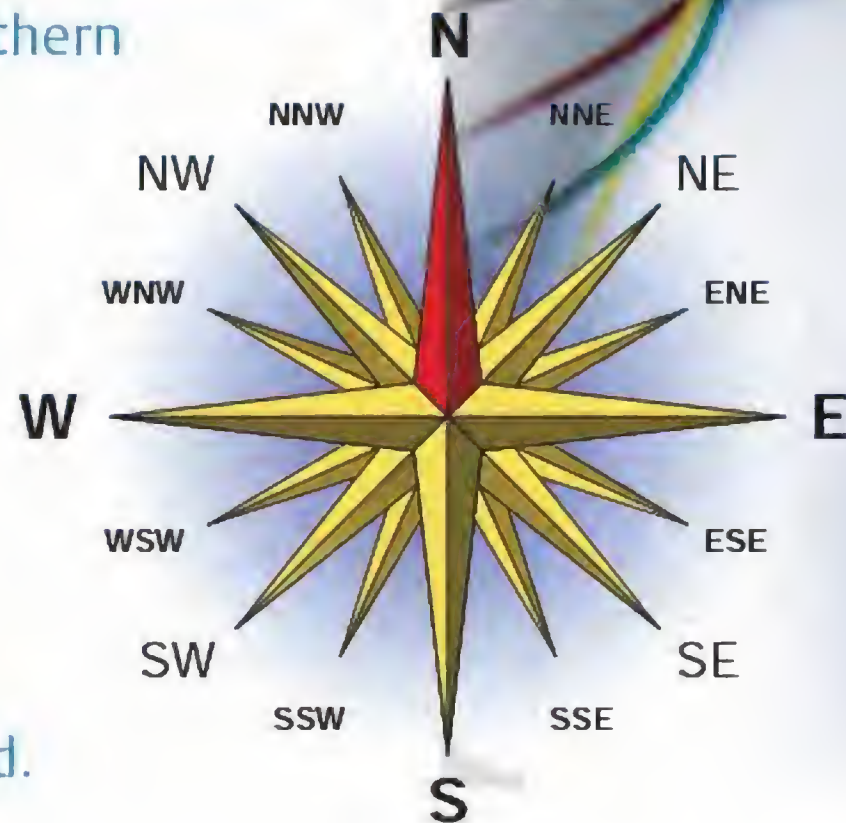
The CMPS03 OEM module [2] makes it possible to calculate the angle between the direction of the Earth's magnetic north pole and the direction in which the sensor is pointing (Figure 1).

To do this, it uses two sensors that are sensitive to the magnetic field of good old Earth. The data is recovered from these sensors and the angle calculated by way of a microcontroller incorporated into the board. A bus connection is also employed, and hence it is possible to recover the value of this angle directly by communication over an I²C bus, either in the form of one byte (0 to 255), or in 16 bits (0 to 3,599); in the latter case, a simple division by ten lets you directly read the value of the angle measured (Figure 2).

ATM18 [3] will be given the task of communicating with the CMPS03 module. The two-wire LCD [4] will be used to display the information obtained.

Figure 1. One application that isn't about to lose its way.

CMPS03 compass module

Don't worry, you won't have to wind metres of wire around a bit if ferrite (Mumetal or Permalloy) that is permeable to the Earth's magnetic field. And no point thinking about a fluxgate system either — that's not the technique this board uses.

The board is based on the use of two Philips KMZ51 magneto-resistive sensors. Their sensitivity is such that the Earth's magnetic field is detected by these two chips, mounted perpendicular to one another. Two sensors are needed, as we need to detect both the North-South and East-West variations. They use Permalloy, a material that's extremely sensitive to magnetic fields, which means the resistance of the sensor changes according to its position with respect to the lines of magnetic flux emitted by the Earth. A Wheatstone bridge on each sensor (Figure 3) makes it possible to accurately determine the value of the vari-
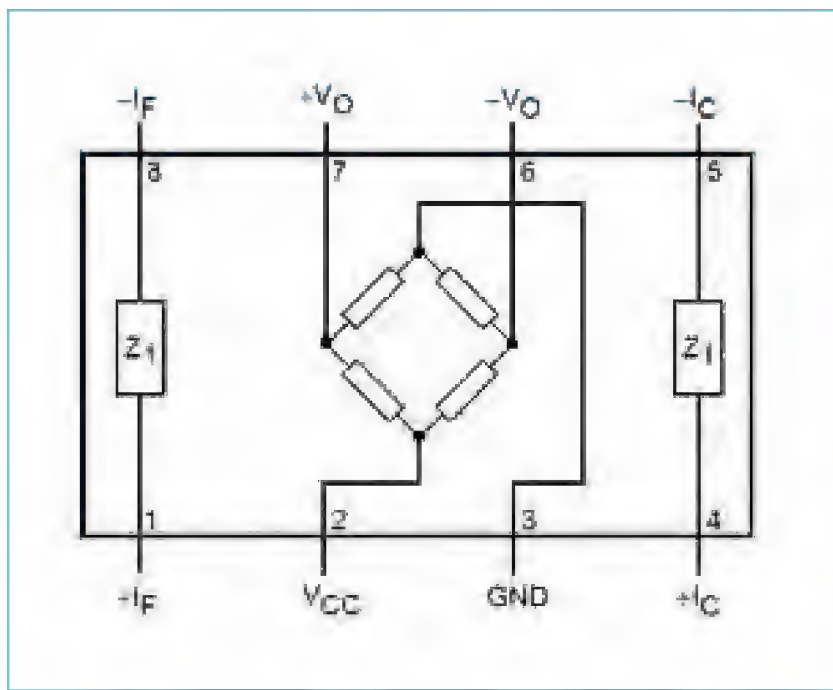
Figure 3. It's all in the KMZ51 chip.

Figure 2. Organized display of the measurement results.

able resistance. The variation in resistance causes a voltage variation which is amplified (LMC6032) and measured (AN0 and AN1 of the PIC) by the CMPS03 sensor electronics. In this way, the magnetic heading is calculated and a microcontroller (PIC18F2321) outputs the result on an I²C compatible bus; all that remains for you to do is to make use of this.

For correct measurement, the module will need to be positioned horizontally with respect to the ground.
Pin 7 on the module makes it possible to correct fluctuations in the signal (jitter) caused by the AC power grid in your home. If this pin is pulled down to ground, the correction is selected for 50 Hz AC; if it is pulled up to 5 V (or left floating), the conversion will be synchronized to a 60 Hz AC supply. The module is supplied factory-calibrated by the manufacturer in the UK (inclination 67°). If you are at a substantially different place on the globe it may be necessary to re-calibrate. This procedure only has to be done once, as the parameters are stored in an EEPROM.

The technical documentation for the module available from the manufacturer's website [5] explains more about this aspect.

Finish first when orienteering!
Start by wiring the circuit by referring to Figure 4, then load the firmware [1], power up, hold the compass horizontal and look for North. How? By turning round holding the circuit. If you are facing exactly North, you'll hear a beep and Cmps03_bearing_byte and Cmps03_bearing_word will take the value 0. These are the two functions, seen in Listing 1, that let you recover the 8- and 16-bit words; they have been written in accordance with the timing diagram provided by the manufacturer (Figure 5).

For example, if you want to recover the two image bytes of the angle calculated by the CMPS03 module, the function cmps03_bearing_word starts by sending a start bit, followed by the module's write address ($C0) and the value of the register (Table 1)

whose contents has to be read (here, it's 2); then you send a new start bit, followed this time by the module read address ($C1). Then the MSbyte is read and its reception acknowledged (ACK), and then to conclude, the command I2crbyte Lo_byte,Nack lets

Listing 1. The registers reveal their contents.

```
Function Cmps03_soft_revision() As Byte
    I2cstart
    I2cwbyte Cmps03_addr_write
    I2cwbyte 0
    I2crepstart
    I2cwbyte Cmps03_addr_read
    I2crbyte Cmps03_soft_revision , Nack
    I2cstop
End Function

Function Cmps03_bearing_byte() As Byte
    I2cstart
    I2cwbyte Cmps03_addr_write
    I2cwbyte 1
    I2crepstart
    I2cwbyte Cmps03_addr_read
    I2crbyte Cmps03_bearing_byte , Nack
    I2cstop
End Function

Function Cmps03_bearing_word() As Word
    Local Hi_byte As Byte
    Local Lo_byte As Byte
    I2cstart
    I2cwbyte Cmps03_addr_write
    I2cwbyte 2
    I2crepstart
    I2cwbyte Cmps03_addr_read
    I2crbyte Hi_byte , Ack
    I2crbyte Lo_byte , Nack
    I2cstop
    Cmps03_bearing_word = 256 * Hi_byte
    Cmps03_bearing_word = Cmps03_bearing_word + Lo_byte
End Function
```
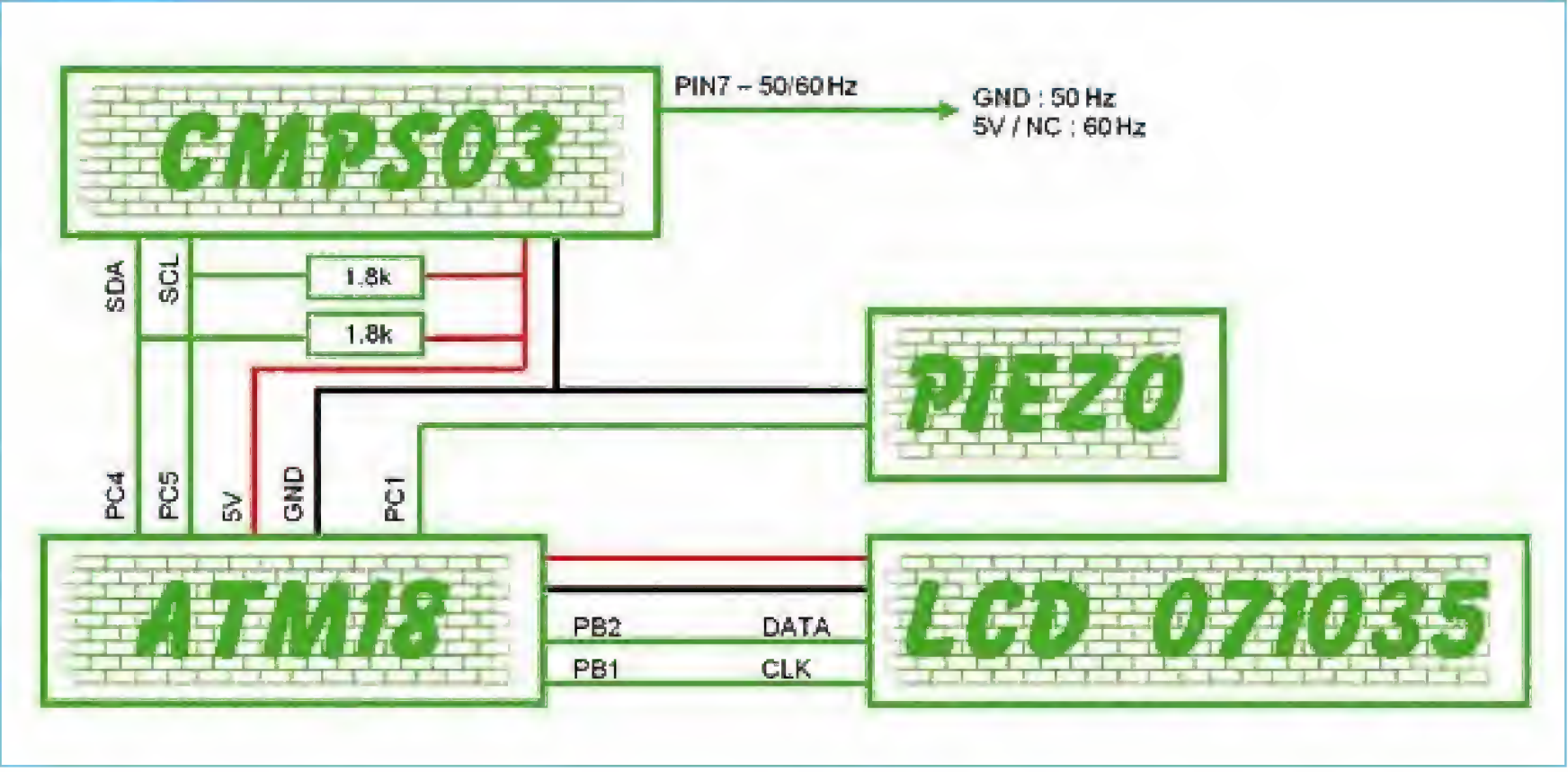
Figure 4. Wiring block diagram.
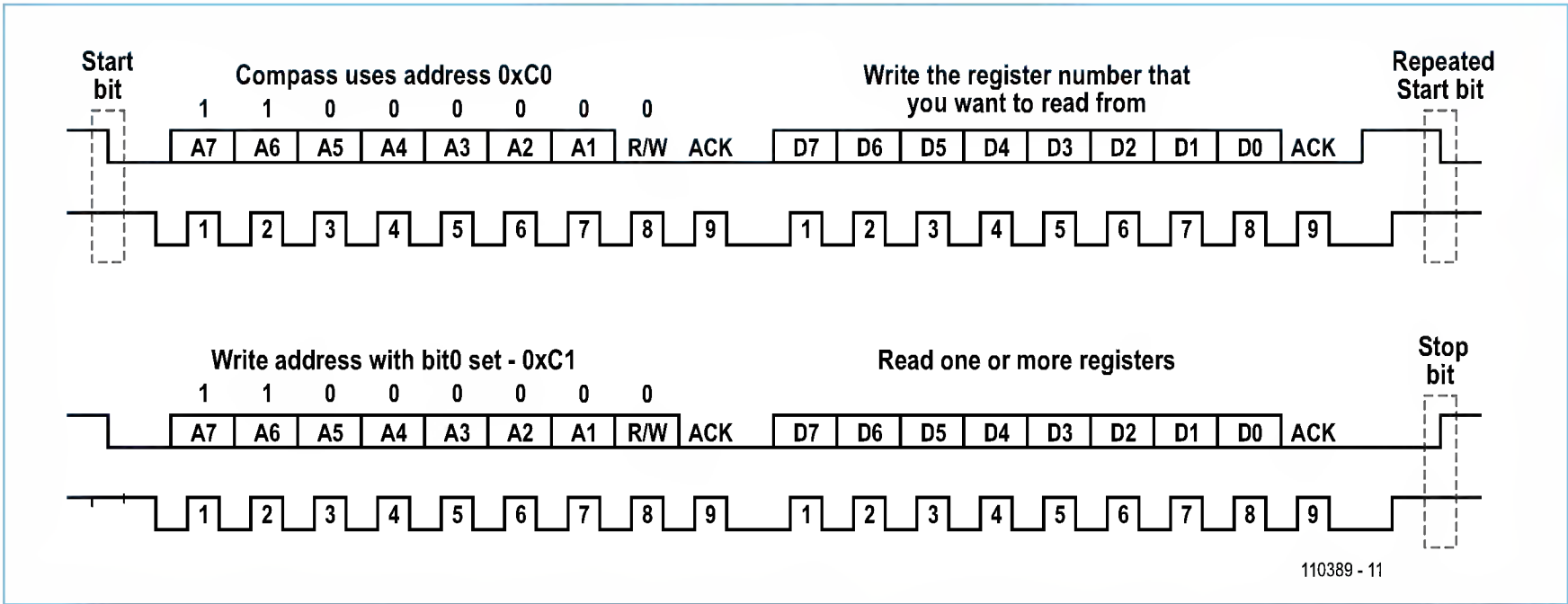
Figure 5. Digital communication with the Robot Electronics CMPS03 module.

you read the value of the LSbyte and store it in the variable Lo_Byte. As this is the last piece of data read, it is not acknowledged.

Graphical representation of the compass points

To help navigation, the 360° are divided into eight parts and the cardinal points are displayed graphically. Figures 6 and 7 show that we are facing North-north-east and South-south-west respectively.

The liquid crystal display is fitted with an HD44780 controller, which looks after character generation and driving the LCD. From the user's point of view, this results in lighter code, as all you have to do is send the controller the instructions, the characters,

and the indications to know where to display them.

The special characters in Figures 6 and 7 are not in the LCD controller's CGROM memory. So they need to be created and saved in CGRAM (Character Generator RAM). In this memory, the user-defined character set starts at address $40 (increments automatically) and contains eight bytes per character. Thus eight custom characters can be stored; these are called for display by sending the ASCII code from 0 to 7 to the LCD.

A 5 × 7 alphanumeric character is formed by seven lines of five dots. There is also an eighth line, which is normally used for positioning the cursor; this will not be modified. BASCOM-AVR provides a tool called LCD Designer (Figures 6 & 7) which

offers help creating custom characters. As this project uses the two-wire LCD, the command line generated automatically by this tool, Deflcdchar [0],31,24,25,27,31,31,31,32, will be entered as a comment, but it does at least enable us to find out the value of the eight bytes corresponding to the character constructed.

For example, in order to create the character seen in Figure 6, we need to send eight bytes each representing one row of dots. The three MSBs are ignored. A pixel is visible if the corresponding bit is set to 1. Listing 2 is an extract from the procedure that lets us store the set of freely-defined characters in CGRAM memory.

Listing 3 reveals part of the contextual procedure designed to position the cursor at the place (X_lcd, Y_lcd) where we want to display the special character. The procedure is called like this: Call Pointing("NNE" , 14 , 4).

So this natural physical phenomenon, the Earth's magnetic field to which we are all of us constantly subjected, has enabled us to produce this project. Nature can make itself very useful!

This very handy sensor can be incorporated into a mobile robot. However, it will be necessary to keep the module well away from any electromagnetic source likely to interfere with the proper operation of the unit.

(110389)

Table 1. CMPS03 registers used in the firmware.	
Register	**Function**
0	Version of firmware in the CMPS03's PIC18F2321
1	Angle in one byte: 0 to 255 for a full circle
2, 3	Angle in one word (two bytes): 0 to 3599 for a full circle. Corresponds to direct reading of the angle from 0 to 359.9 degrees.
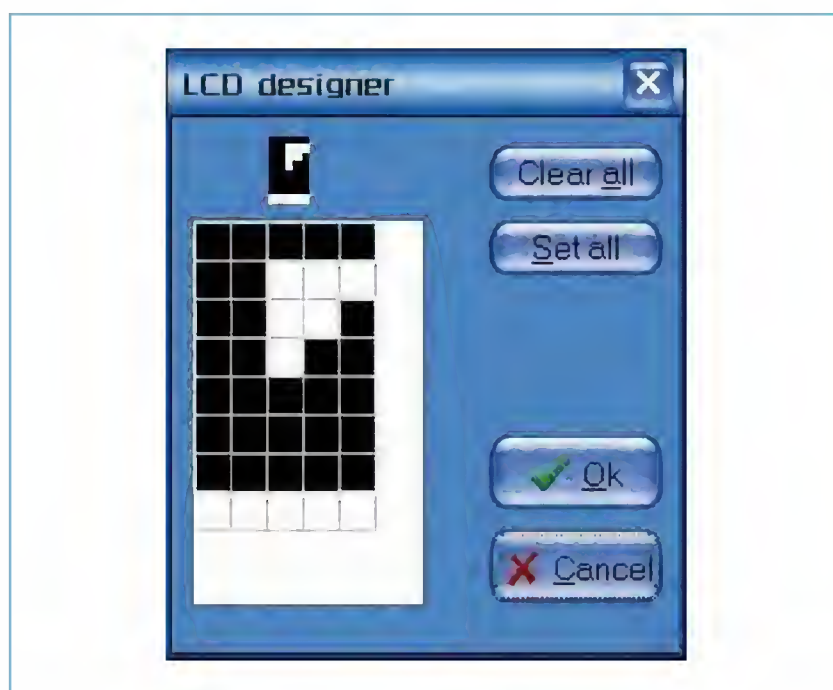
Figure 6. The LCD Designer tool from BASCOM-AVR...
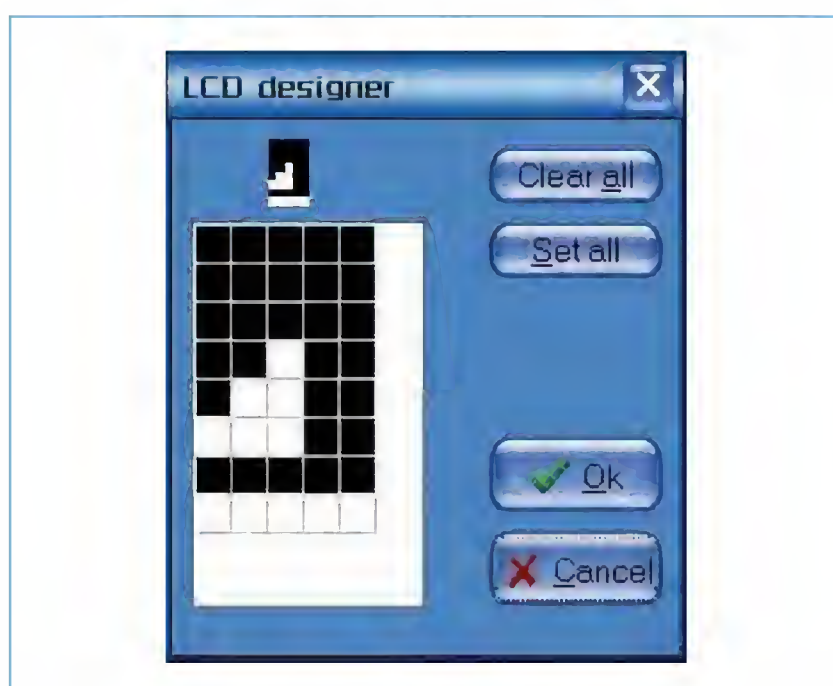
Figure 7. ...for "fine-tuning" your characters.

Internet Links

[1] www.elektor.com/110389

[2] www.robot-electronics.co.uk/

[3] www.elektor.com/atm18

[4] www.elektor.com/071035

[5] www.robot-electronics.co.uk/htm/cmp-s3tech.htm

Listing 2. Storing custom characters.

```
'ADRESSES CG-RAM A PROGRAMMER
'CHAR1 : $40 à $47
'CHAR2 : $48 à $4F
'CHAR3 : $50 à $57
'...
Sub Lcd_custom_char()
    Rs = 0
    Waitms 20
    Lcd_write_byte &H40         'adresse de base CGRAM
    Rs = 1                      'envoi des données
    Waitms 20
'---------------CHAR1_NNE
'Deflcdchar [0],31,24,25,27,31,31,31,32
    Lcd_write_byte 31          '$40
    Lcd_write_byte 24          '$41
    Lcd_write_byte 25          '.......
    Lcd_write_byte 27
    Lcd_write_byte 31
    Lcd_write_byte 31
    Lcd_write_byte 31
    Lcd_write_byte 32
'---------------CHAR2_ENE
    Lcd_write_byte 31          '$48
    Lcd_write_byte 30          '$49
    Lcd_write_byte 28          '.......
    Lcd_write_byte 24
    Lcd_write_byte 31
    Lcd_write_byte 31
    Lcd_write_byte 31
    Lcd_write_byte 32
'---------------CHAR3_ESE
    Lcd_write_byte 31          '$50
..
..
..
End Sub
```

Listing 3. Outputting the custom characters.

```
Sub Pointing(byval Direction As String , Byval X_lcd As Byte , Byval Y_lcd As Byte)
Lcd_pos X_lcd , Y_lcd
Select Case Direction
    Case « NNE »
        Rs = 1         ' envoi de données
        Waitms 20
        Lcd_write_byte &H00
    Case « ENE »
        Rs = 1         ' envoi de données
        Waitms 20
        Lcd_write_byte &H01
..
..
End Select
End Sub
```

J²B: Universal MMI Module using ARM Cortex-M3

Let's stop reinventing the wheel!

By Clemens Valens (Elektor France Editorial)
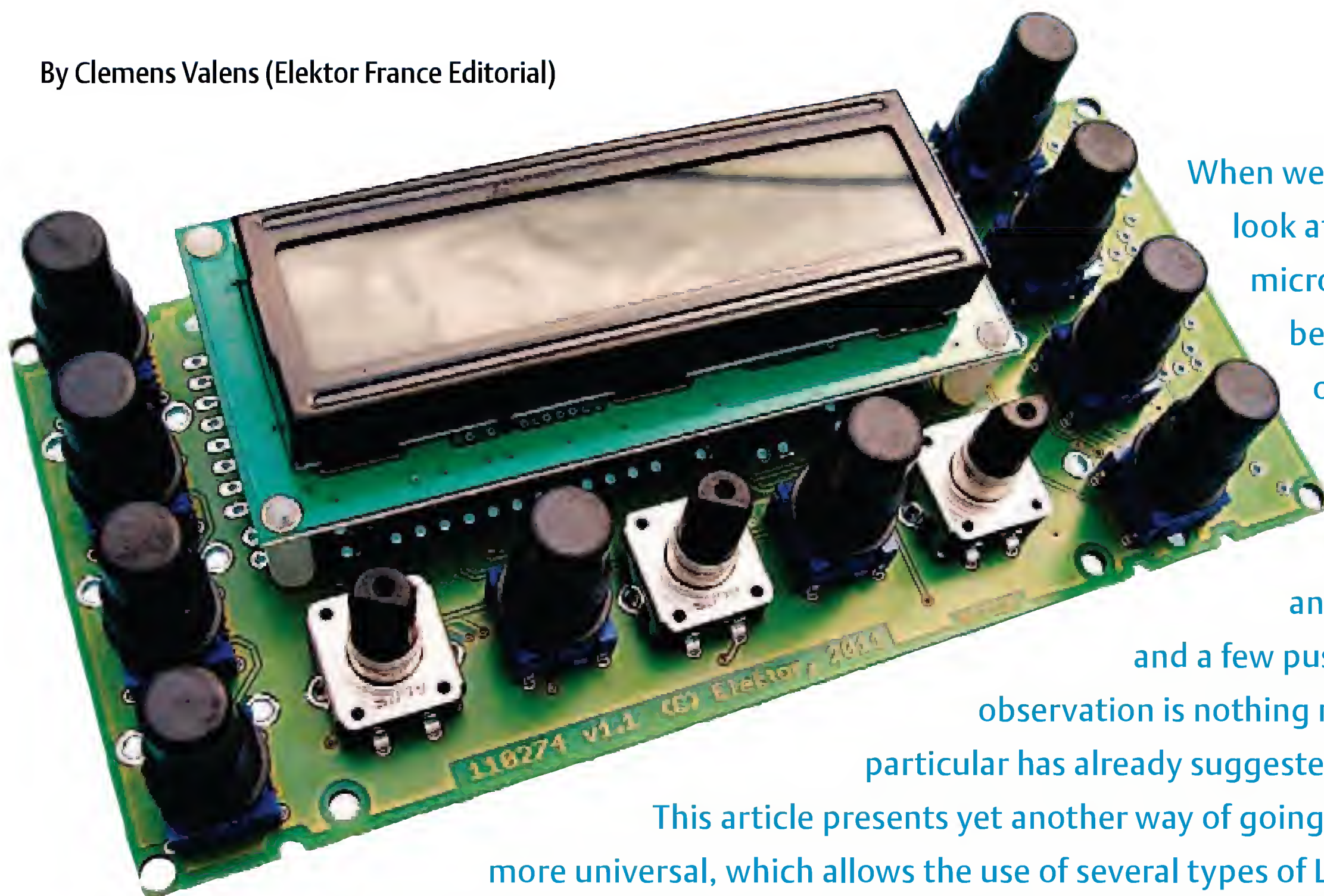
When we take a closer look at circuits using a microcontroller, it can be seen that in 75 % of cases the basic circuit is virtually identical: a microcontroller, an LCD display, and a few push-buttons. This observation is nothing new, and Elektor in particular has already suggested several solutions. This article presents yet another way of going about it, a little bit more universal, which allows the use of several types of LCD and a variable number of buttons. And thanks to its up-to-the-minute LPC1343 ARM Cortex-M3 processor, this board is extra powerful and amazingly easy to use.

The type of displays usually used in amateur projects generally have either 2 lines of 16 characters (2×16), or 4 lines of 20 characters (4×20). There are rarely more than four buttons, but rotary encoders are increasingly being seen. For circuits using a 2×16 LCD, the buttons are often below the display; for the 4×20 displays, they are more likely to be mounted to the side. The position of the buttons depends on the application and the user; a right-handed person tends to position them differently from a left-handed person. A universal solution must take all this into account and allow a free choice of the LCD and the position of the buttons.

Even though I'd originally had the idea for this circuit several years ago, I started actually building it quite recently, when for the umpteenth time I needed to add an MMI to a circuit that didn't have enough I/Os free; so I needed an additional port extender. Since NXP brought out their 32-bit ARM Cortex-M3 and M0 microcontrollers, cheaper than port extension devices, an ingenious and inexpensive solution is now possible. Particularly interesting for amateurs, this is the LPC1343, currently the easiest microcontroller to program. No need for a programmer or an RS-232/USB adaptor — this microcontroller is presented quite simply as a USB stick onto which all

you have to do is copy the software (only works under Windows; using Linux or MAC OS, you have to use a serial link or a special programmer).

Using a microcontroller like this as a port extender, we get USB, I²C, and SPI ports or a UART for communicating with the application. If we note that in most applications managing the display and the keyboard takes up easily 80 % of the software, we can likewise envisage having the whole application run by the microcontroller — especially when we have the computational power of a 32-bit processor available. So, instead of adding a port extender to an applica-

Figure 1. Block diagram of the J2B board.

tion, we can add an application to a port extender.

Since the microcontroller is only available in an SMD package, I decided to use only SMD parts throughout (with the exception of the connectors and keyswitches). In the relatively confined space, cluttered with connectors, this allowed other functions to be added, like powering by rechargeable batteries or primary cells, and a charger for Lipo batteries. This means that the board is also suitable for mobile applications.

And to round everything off nicely, the board dimensions are suitable for a standard, cheap case, for a neat, robust finish for your project.

Specifications

- Suitable for 2×16, 4×16, and 4×20 LCD displays using a standard 14- or 16-pin connector (the latter if backlight is included), software-controlled backlight;
- 5×6 matrix keypad for a maximum of 12 keyswitches or nine rotary encoders with built-in push-button (equivalent to 27 pushbuttons!) or a mixed configuration;
- Buzzer;
- One LED;
- Power via USB, external 5 V supply, primary cells (0.9–4.5 V) or Lipo rechargeable battery;
- 5 V and 3.3 V regulators, software on/off control possible;
- Lipo battery charger, plus software measurement of battery level;
- 32-bit, 48-pin LPC1343 microcontroller with 32 KB Flash memory, 8 KB RAM and numerous peripherals like USB, I²C, SPI, MLI, UART, and counters;
- Compatible with the free LPCXpresso [2, 3, 4] and CooCox [7] IDEs;
- Compatible with the LPC-Link and CooCox programmers/debuggers;
- Extension connectors: almost all the microcontroller pins are available on a connector or a pad;
- Splittable! Certain unused parts can be cut off; detachable mini 4-key pad or a maximum of 3 rotary encoders;
- Dimensions adapted to Type 26160000 case from Bopla;
- Open source software and hardware.

Operation

Microcontroller

Let's start with the board's brain, the microcontroller. An oscillator is necessary to make it work, and so three options present themselves: the internal RC oscillator (IRC), an external oscillator, or an external clock (which we are not using here). The microcontroller starts up on its internal oscillator. For applications requiring a more accurate clock, provision has been made to allow a crystal oscillator to be fitted. The microcontroller's data sheet advises values of 18 pF or 39 pF for C14 and C15, depending on the crystal used. However, most crystals ought to be able to operate with either of these values.

The logic levels on PIO0.1 and PIO0.3 select the microcontroller's start-up mode after a reset. If PIO0.1 is low, the microcontroller will first run its bootloader (ISP mode), otherwise the user program will be run. The LPC1343 offers two ISP modes: by USB stick (PIO0.3 high) and by serial port (PIO0.3 low). Resistors R3 and R13 let us select the ISP mode. In theory, the function of R3 is also fulfilled by the USB voltage detection circuit, but fit it all the same in order to avoid any possible indeterminate situations, if ever the USB cable should not be connected (correctly). Fit R13 instead of R3 if you prefer to program the chip via a serial link.

Figure 2. Circuit diagram. The construction file (Eagle files, component list, software, etc.) is available on [1].

110274 - 11

Note that there is a bug in the LPC1343's built-in USB driver, which means you have to wait around 30 s before Windows will detect the "stick" the first time you connect it for a programming session. After that, the detection takes place normally for as long as the microcontroller remains powered.

The microcontroller can also be programmed via an SWD (Serial Wire Debug) port, a sort of serial JTAG. Connector K3 has been provided for this purpose, wired in such a way as to be compatible with the LPC-Link used by the free LPCXpresso programming environment. The LED (D34), connected to PIO0.7, is also LPCXpresso compatible, which means that the little test program LPCXpresso1343_blinky included in the IDE ought to work without any modification.

Keypad

The keypad consists in principle of twelve switches, four to the left of the display, four to the right, and four below (with a 2 × 16 display). The idea is to also be able to use rotary encoders in place of switches; an encoder of this type with built-in push-button is equivalent to three switches. Three encoders take up the same space as four switches. To leave as much freedom as possible in positioning the switches and encoders, I decided to allow up to nine encoders (in point of fact, I have an application that needs that many rotary encoders). So nine encoders with built-in buttons correspond to 27 keys, thus a 5 × 6 matrix is needed to connect all the keys using a minimum of I/Os. The matrix is wired in such a way that if you are only using one of the three groups of four keys (left, right, or bottom), only four I/Os are needed for scanning these four keys. The I/Os freed up in this way are available for some other function. JP1 makes it possible to optimize the position of S5 in the matrix.

I have used diodes to avoid the problem of phantom keys if several keys are pressed at once – a situation that can easily arise when you're turning two encoders at the same time. The diodes are in a SOD323 package, allowing them to be replaced by resistors, in case you might want, for example, to fit one or more LEDs in place of certain keys.
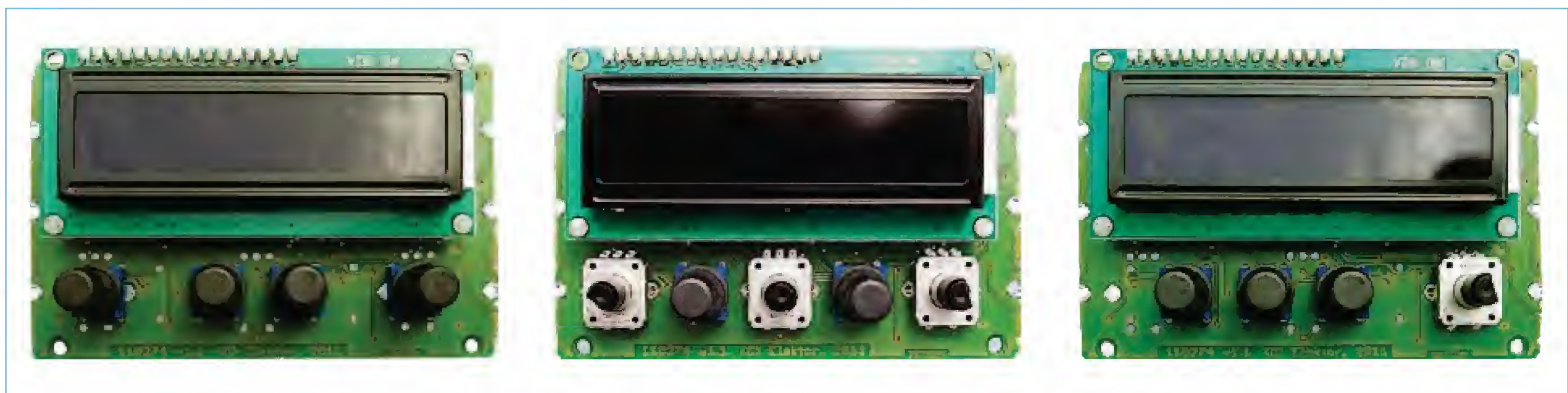
Figure 3. Different variations on the 2×16 LCD theme. Note that the keyswitches can be offset.

The main function of resistors R38, R39, R43, and R44 is to enable the board to be turned on using one of the buttons S1, S5, S8, or S12. If one of these keys is fitted with its corresponding bridge (R32, R35, R37, or R42), the resistors provide the voltage needed to turn on T5 when the key is pressed. The circuit can be turned on this way if powered by battery. The function of R47 is to avoid a wiring asymmetry in the matrix; it is not used for an on/off button.

All the matrix lines have a current-limiting resistor (R15–R25). Its value is not very important, just so long as it doesn't impede the detection of the key presses.

Finally, note that two footprint are available on the circuit board for S9–S11 and S19–S21.

Display

The board offers the possibility of fitting a 2 line by 16 character ('2×16') display, a 4 line by 20 character (4 × 20) type, and even a rather less common 4 line by 16 character (4×16) type — just so long as the display has a standard connector with a single row of 14 or 16 contacts at the top left. This configuration is very common. Versions of the 4×20 displays also exist with two of these connectors, allowing them to be fitted either way up.

This type of display is normally powered at 5 V, which is not really a problem when the board is connected to an external 5 V supply (USB for example), but to allow battery powering, I've added a voltage booster that steps the voltage up to 5 V on the board.

T3 offers the possibility of turning off the power to the LCD, which is handy for resetting the display or for limiting the circuit's overall power consumption. For this same reason, the backlight is also driven by a transistor (T1), with R1 to limit the cur-

rent. This resistor is in the 1206 format for better heat dissipation. Its value is not critical and depends on the brightness required.

Buzzer

A standard 12-mm diameter piezo buzzer (with leads on a 6.5 mm pitch) offers the possibility of producing sounds and alarms.
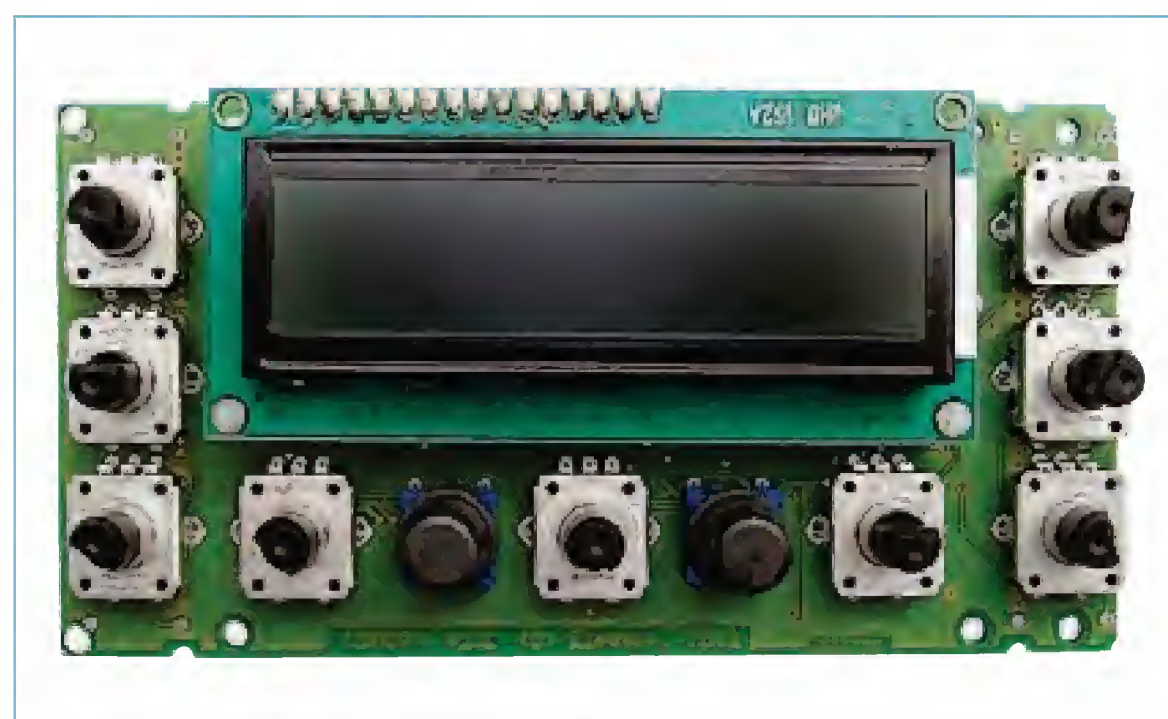
Figure 4. The maximum: nine rotary encoders and two buttons — a total of 29 switches!

It is driven by T2. Resistor R36 avoids spurious squealing when PIO2.8 is configured as an input during programming of the microcontroller. R41 is not necessarily useful for a buzzer, but does allow T2 to be used to drive something else like a relay or an LED. Like R1, it too is in a 1206 package.

Powering

The board's main power rail (V+) is 5 V. This voltage is needed for the display, as the microcontroller itself is powered at 3.3 V and everything is connected to the microcontroller. The 3.3 V is derived from the 5 V rail by way of a low voltage drop regulator IC3. R48 allows this rail to be disconnected so the power supply can be tested without any risk to the microcontroller.

The 5 V can come from three sources:

an external supply connected to K7, the USB port (K4), or IC2. The latter is a voltage booster which accepts an input voltage from 0.9–4.5 V and hence offers the possibility of powering the board from a rechargeable battery or one or more individual 1.5 V cells (K5). The microcontroller does not use much power (60 mW in normal operating mode, clocked at 72 MHz) and a single 1.5 V cell is enough for long hours of operation. For information, the microcontroller operates with a supply voltage between 2 and 3.6 V, so a 3.3 V regulator is not indispensable, given that a 3 V lithium battery or primary cells is amply sufficient. In the event of using a Lipo battery (< 4.5 V) — for example, out of a mobile phone — a battery charger (IC4) can be fitted to the board, powered by USB (or K7).

The microcontroller is capable of measuring the battery level by way of resistors R11 and R12.

If the board is powered from an external supply only, there's no need to fit all of the power supply. In this event, fit just D32, C3, C4, C5, IC3, and R48. Note that in this case, the software-driven on/off function will not be available.

USB and extensions

Let's finish our description of the circuit by the extension and communication ports.

K4 is a mini USB connector mounted on the board. It is used above all during the development phase for programming the microcontroller in ISP mode via a USB stick. For a finished application, this connector is probably not in the right place, which is why there is connector K7 which makes it possible to remote the USB connector. T4 is driven by the microcontroller to indicate to the computer that a USB peripheral has been connected.
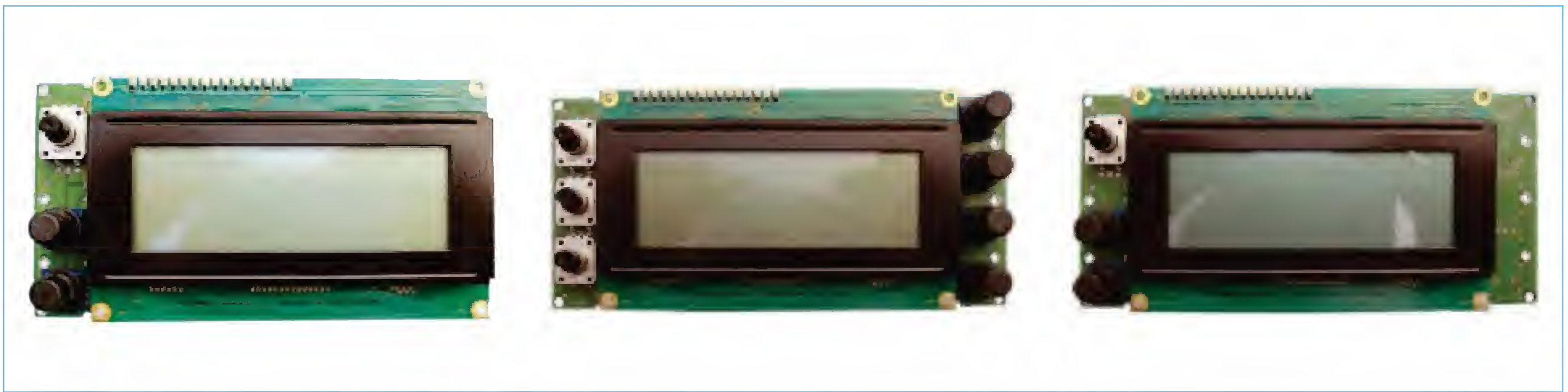
Figure 5. Plenty of possibilities too with a 4×20 character display.

Virtually all the microcontroller I/Os are directly connected to extension connectors K1, K2, and K6. We can also use K8, normally used for connecting the display. A few I/Os are available only via a transistor (PIO0.6, PIO2.8, PIO2.9, and PIO2.10), PIO0.1 is available on JP2 and PIO0.7 on D34. In all cases, each I/O is connected to a pad.

K1 and K2 carry the signals from the communication ports like the I²C, SPI, and UART. Here too are the I/Os not used by the matrix keypad or the LCD. For those who have a USB/TTL adaptor cable from FTDI (Elektor ref. 080213 [8]), you will be interested to know that K1 is compatible with this cable.

Note that port PIO1.4 has a special function in the microcontroller's 'deep power-down' mode, which is why it has a pull-up resistor R10. Note too that two pull-up resistors (R4 and R5) can be fitted for the I²C port.

K6 offers access to most of the ports used for the keypad. A certain number of these signals are also available on K8 and K9, but these two connectors are intended for instances where one might want to split off a mini 4-key keypad for mounting elsewhere.

Construction

None of the SMD devices on the board are too difficult to solder. The microcontroller could prove trickiest, but armed with a bit of desolder braid, it's easy enough to get rid of any surplus solder.

We find more difficulty with the inductors, as they're not always easy to source. Hence I opted for mixed sizes, more or less suited to the types from Coilcraft (try their excellent sample service! [9]), from Coiltronics (distributed by Farnell, for example) and through-hole types on a 3.5 mm pitch.

Different types from other manufacturers might well also be suitable.

In principle, the buzzer, LED, connector K8, and all the keyswitches are fitted on the solder side of the board.

The switches are modular types, consisting of a body onto which can be clipped caps of different colours, shapes, and sizes. Thus
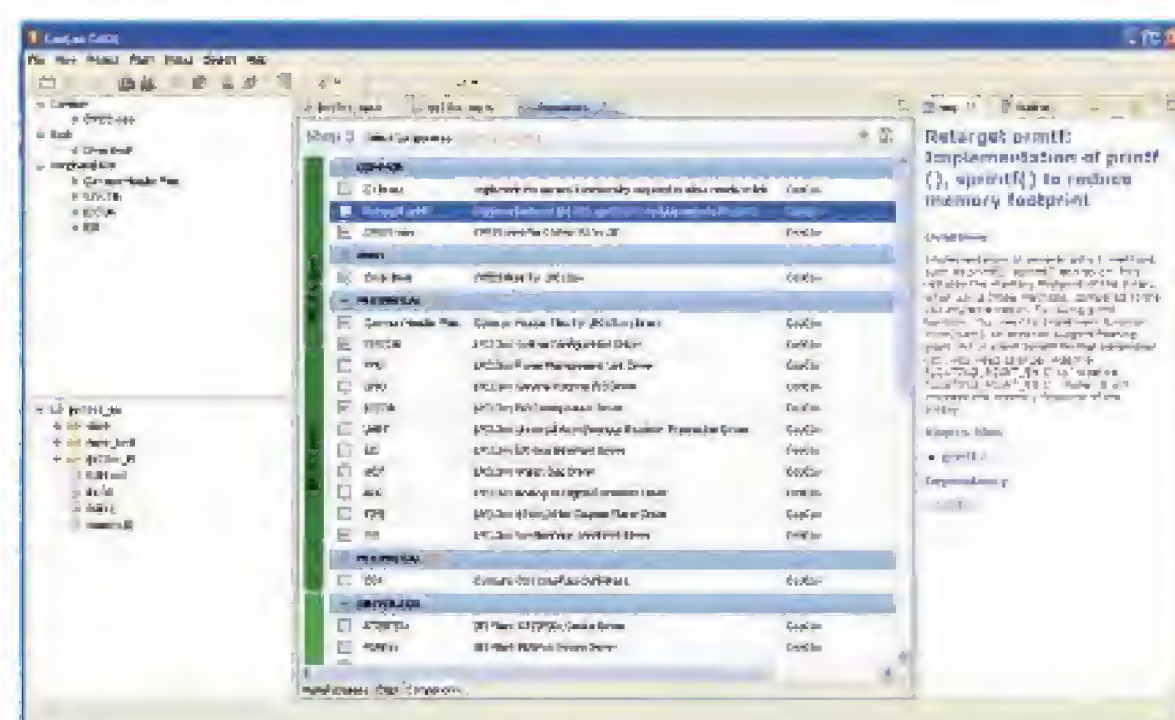
Figure 6. CooCox CoIDE [7], a powerful free IDE for ARM Cortex-M0 and M3 processors from several manufacturers.

each constructor can use keyswitches suitable for their application and their taste. Depending on the configuration of the keyswitches and display adopted, the board can be split so as to make it more compact. Predrilled dotted lines make it easier to split the board to suit the dimensions of the chosen case. The whole board will fit into a case from Bopla (ref. 26160000). Using a 2×16 display and four buttons below, the board is smaller than a 4×20 display. Because of the removable keypad, the board with a 4×20 display is almost the same size as the 4×20 display itself.

Implementation

The board described in this article is compatible with the LPCXpresso from NXP,

Embedded Artists, and Code Red [2, 3, 4] integrated development environments (IDE). This free IDE is without restrictions for the LPC1343 and is supplied with numerous examples and several libraries. The IDE is complemented by a number of LPCXpresso boards that include a programmer/debugger named LPC-Link, which is to a greater or lesser extent detachable, and a variable microcontroller part. There is also an LPCXpresso board based on the LPC1343 and our board is compatible with this.

If you have an LPCXpresso board, cut off the LPC-Link part and consult the article "Getting Started with your Free LPCXpresso Board" published in the 2011 double issue [5]. In this article, replace all references to the "1114" by "1343" and connect the LPC-Link to K3 on our board. If you have fitted the LED, all you have to do now is follow the instructions to make it flash.

If you don't have the LPCXpresso board, you can still use the IDE by using the microcontroller's ISP via USB stick mode. To do this, all you have to do is configure the IDE to produce an executable file in the right format.

To do this, in the menu, click on *Project*, then *Properties*. Click on the + in front of *C/C++ Build* and select *Settings*, then click the *Build Steps* tab. In the *Post-build steps* group, enter the following into the *Command* box:

```
arm-none-eabi-size
${BuildArtifactFileName}; arm-
none-eabi-objcopy -O ihex
${BuildArtifactFileName}
${BuildArtifactFileBaseName}.
hex; arm-none-eabi-
objcopy -O binary
${BuildArtifactFileName}
firmware.bin; checksum
firmware.bin;
```

Source: Taco Zip 3 – Heavy kunst (1992) by Fritzgerald, Jakketoe & Cromheecke. Made available free of charge by Luc Cromheecke. Translated from the Dutch by Elektor.

This is in fact a series of commands, some of which are already present (but watch out for the '#' characters which must be deleted). What's important here is that you end up with the file produced called **firmware.bin** and that the checksum is added to it (otherwise the executable won't be recognized by the microcontroller).
End by clicking on 'OK'.

From now on, the IDE will produce an executable that you'll be able to copy onto the USB stick (which always contains just a single file named firmware.bin). Don't forget about the bug in the LPC1343's USB driver which means that it sometimes takes 30 or more seconds before the stick is recognised.
Apart from LPCXpresso, there are also other free possibilities. To start with, there is the microBuilder website [6] which describes how to prepare a programming environment for the LPC1343 bases on Yagarto. This site also offers a library to help you get

started quickly with the microcontroller. I made use of this library in developing my test program [1].
Closer to LPCXpresso, there is CooCox CoIDE [7]. Just like LPCXpresso, this IDE is based on Eclipse and it includes numerous drivers for the microcontroller peripherals and lots more besides, like the CoOS RTOS. A programming and debugging probe

is also available, based on the LPC1343. You can buy it (watch out for the carriage charges!), but you can also build it yourself, as it's an open-hardware project. Here, the microcontroller's ISP via USB stick option avoids getting into a chicken-and-egg situation. Note that neither CoIDE nor LPCXpresso calculates the checksum for the executable automatically.

Who's afraid of 32 bits?
I'm aware that many of our readers don't feel comfortable with 32-bit microcontrollers, even though nowadays they are easier to implement than 8-bit microcontrollers, and what's more, cheaper. For these readers, in a future (the next?) issue I'll be presenting a board similar to the one described here, but simplified and based on an 8-bit AVR microcontroller. It won't use SMD components and — the icing on the cake — it will be Arduino, Mikroelektronika, and BASCOM-AVR compatible.

(110274-I)

Internet Links
[1] www.elektor.com/110274

[2] http://ics.nxp.com/lpcxpresso/

[3] www.embeddedartists.com/products/lpcxpresso/

[4] http://lpcxpresso.code-red-tech.com/LPCXpresso/Home

[5] www.elektor.com/110448

[6] www.microbuilder.eu/projects/
LPC1343ReferenceDesign/LPC1343Toolchain.aspx

[7] www.coocox.org/

[8] www.elektor.com/080213

[9] www.coilcraft.com

[10] http://elektorembedded.blogspot.com

Modifications to the microBuilder library

As mentioned in this article, I based my test application on the open source library for LPC1343 from microBuilder [6]. My program, also open source, is available from [1]. I've had to modify the microBuilder library, as it had a few drawbacks. So it's no longer possible to use the original library. Here are the main modifications I've made:

- gpioSetValue: now uses the bit masked version of GPIOnDATA to avoid conflicts when this function is called from an interrupt service routine;
- UART: the uartRxBuffer buffer has been replaced by a more universal buffer called uartBuffer. Interrupt-controlled transmission has been added;
- SPI: numerous modifications have been made;
- cmd.c: a mechanism was lacking for avoiding the message buffer memory's overflowing.

A ghost in the machine

By Dr. Thomas Scherer (Germany)

It must have been about a year ago when I first noticed it. I plugged my electric corkscrew (what, doesn't every engineer carry one in their toolbox?) into its charging station as usual but this time the charging LED began flashing on and off. Odd, I thought, it's never done that before; the charger circuit must be more sophisticated than I'd imagined. The irregular flashing continued and the LED started to change colour from green to yellow/green. Curiosity got the better of me, armed with a screwdriver I set about dismantling the unit. Inside I couldn't see any sign of a chip, transistor or even a capacitor or coil in fact there was just an LED and a series resistor.

The circuit (shown here) really was as dumb as I had first thought. The circuit diagram shows that charging current for the four batteries flows direct from the 9 V AC power adapter through the LED, limited by a series resistor. I carefully de-soldered the LED and connected it to a bench power supply; sure enough it started blinking again, I turned up the current and the blink rate increased until at about 70 mA it gave up the ghost. Although a little surprised, I replaced the charger LED and increased the value of the series resistor to 470 Ω. For sure, this would produce a continuous charge current of around 5 mA which should be enough to keep the batteries charged. I had almost forgotten the whole episode until recently when my wife called me into our living room to point out that one of the red power LEDs in a lamp on to of our cabinet had started flashing. The spirit of the dying LED had made an unwelcome return. In this case the irregular flashing only lasted for about 15 minutes before it departed for good. Once again the circuit was just an LED and a series resistor. The photo of the dead
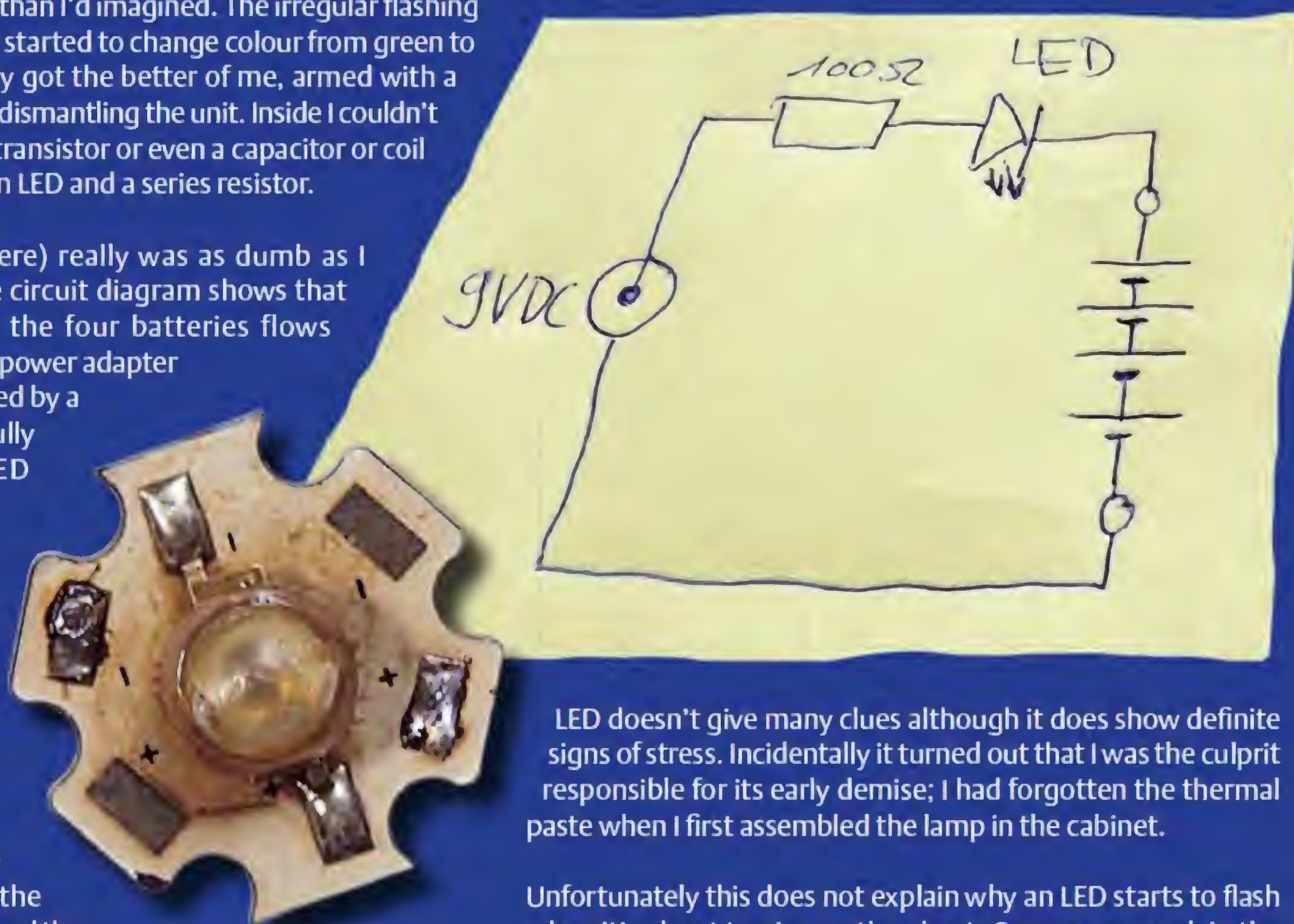
LED doesn't give many clues although it does show definite signs of stress. Incidentally it turned out that I was the culprit responsible for its early demise; I had forgotten the thermal paste when I first assembled the lamp in the cabinet.

Unfortunately this does not explain why an LED starts to flash when it's about to give up the ghost. Can anyone explain the mechanism at work here? I'm intrigued. Send in your answers to Jan at editor@elektor.com; we will publish the best.

(110459)

E-LABs INSIDE

Alibaba

By Thijs Beckers (Elektor Netherlands Editorial)

Don't you think that the world seems to get smaller every day? Whereas in the days of Jules Verne it was unthinkable to travel round the world in 80 days, nowadays it's considered on the slow side to do it in 80 hours. I remember that when I was a child we occasionally visited a 'distant' family. In this case, 'distant' meant a 160-mile trip in our Mini Cooper Station, which was quite an experience in those days!

These days we 'nip' over to the stock exchange in London or New York and holidays should be taken in a different country and preferably on another continent. Most of the electronic parts and equipment are no longer manufactured in Europe or the US, but are made in one of the low-wage countries and subsequently transported halfway round the world.

The latter affects one of the jobs that you also have to do as a designer, which is the search for suppliers of the components you plan to use in the circuit you're designing. During a recent search I stumbled across a website called alibaba.com. Perhaps you heard of it already? It is a type of 'portal' for worldwide traders. You can find just about anything there: from rice to cars, from chemicals to golf accessories. And last but not least, electronic parts from China. You do understand why I ended up at this site?

A very useful feature of this site is that it lets you chat online with an employee of the relevant supplier. This way you can ask questions directly about the product and discuss the delivery options. I put this into practice for a future project and made an agreement with a Chinese supplier regarding component samples and the supply to Elektor readers around the world. I won't yet give away for which component this was, since the project is still in its early design stages. But if the project proves successful with the intended components, then the availability of the parts will at least be guaranteed.

Since the Alibaba website is used by suppliers from all over the world you do have to take account of the time-difference and any possible national holidays at the supplier's location before you make contact with them. I have to admit that the whole experience of this site has left a positive impression on me. Doing business over long distances has never been as easy as it is now. The age of information technology certainly has made life easier in this respect.

The world obviously hasn't literally become smaller, but distances no longer seem to matter as much, which is very useful for designers. We'll just have to hope that the Forty Thieves don't make an appearance...

(110549)

Perfect pizzas

By Thijs Beckers / Jan Visser
Elektor Netherlands Editorial / Elektor Labs)

No, we haven't made any drastic changes to our area of interest and plunged ourselves into the writing of recipes or cooking courses. Neither are we reporting on our experiences with the local bistros and pizzerias while we were holidaying in Italy! No, we are talking here about our new, high-tech SMD oven.

In another article in this edition you have already had the opportunity to study the specifications of this swanky baking machine. What you haven't been able to read, are the types of tests that the new machine was subjected to by our lab

colleague Jan Visser. By far the nicest, or better: the tastiest, was the heating up of lunch, which was, especially for this purpose, unearthed from the freezer at the supermarket.

To be able to heat the pizza just perfect, Jan, after some trial and error (such punishment!), was able to establish the optimal curve for heating his lunch. Never again will you have to suffer charred pizzas with black edges and shrivelled up mushrooms. And this makes nice change from the usual smoke of molten solder and scorched flux that permeates the lab.

Jan: "To ensure that the pizza is heated as fast and as uniformly as possible, we first pre-heat the oven so that it is already warmed up. We do this by allowing the oven to heat up without

E-LABs INSIDE

anything in it, that is, by running it through a so-called short curve. While the oven is going through this heating cycle we use this opportunity to retrieve the pizza from the freezer and unwrap it.

Because of the progressively wound spirals of the heating elements, which, in addition, are also positioned in specifically selected places, the oven has an extremely favourable and balanced temperature distribution, so that even large surfaces are heated uniformly.

When placing the pizza in the oven, you need to ensure that the pizza is placed neatly in the middle of the oven so that the heating process will progress as uniformly as possible. The upper and lower temperature sensors are not allowed to 'see' each other. So therefore put the pizza exactly between them. With mini-pizzas this can be a bit of a challenge.

Since our pizza has a preparation time of 8 minutes at 220 degrees Celsius, we adjusted the curve in our program so that this temperature is reached after 1 minute and is then held constant for exactly 8 minutes. The external temperature sensor can in this case be used to measure the outside of the crust, so that we can be ensured of obtaining a "crispy crust".

When shopping for a pizza make sure that you buy a pizza with a short preparation time, somewhere around 8 minutes. Pizzas with a longer preparation time will need a different curve, of course. Pizzas which are thicker than 2.5 centimetres must be avoided, because these would touch the top temperature sensor, and as a consequence a perfect result will not be obtained.

There is no point in making the leading edge of the pizza-temperature curve very steep, because the pizza itself needs time to warm up and can't follow such a curve quickly enough. Obviously, the supplied PCB holders are not used and it is a good idea to put baking paper underneath the pizza to prevent leaks. After 9 minutes our Italian delicacy is ready and the door of the oven opens automatically. The forced cooling period that the SMD oven normally goes through when soldering printed circuit boards must be avoided, of course, and therefore we immediately have to get busy with the pizza cutter to share the hot pieces of pizza with our hungry colleagues, who in the meantime have been attracted by the aroma.

Have you become hungry? Our pizza-curve is available as a download for everyone who would like to experiment with it, you can get it at www.elektor.nl/110537.
Enjoy your meal!

(110537-I)

Problems under pressure

By Luc Lemmens & Thijs Beckers (Elektor Labs)

Here at Elektor Labs, while testing a prototype of the USB Weather Station we came across a strange phenomenon. When measuring the relative humidity, the frequency generated by the sensor is subjected to a calculation in order to arrive at the displayed value. The prototype found the environment to be extremely humid, because we read a value of more than 150% on the display! Although the basement of our castle, where the lab is located, is quite humid, 150% is, of course, not possible. The strange thing was that our prototype used to work correctly (or at the least: indicated no value above 100% — it actually displayed a value that was too low). There had to be something else going on.

The prototype we received from the author was already on its way back to him, because that (also) indicated a value that was

too low and the author was keen to investigate that further. So a comparison between the two assembled circuits was a bit difficult at this time. In addition, the deadline for this issue was breathing down our neck and there was not enough time to

have the author return his prototype to us again.

The calculation in the software was checked once again, the PCB was checked (for the umpteenth time) for potential errors and short circuits. But everything appeared to be all right. The only other thing Luc could come up with was that either the calibration value (stored in the EEPROM of the sensor) was corrupted, which is very unlikely since a second sensor module gave exactly the same error, or that the software was reading the calibration value from an incorrect address.

At the time of this edition going to press the problem hasn't been solved, but we trust that this will definitely be the case by the time you read this magazine and that the software-download for this article will be 100% functional.

(110383)-I

Small pitfalls

By Thijs Beckers & Ton Giesberts (Elektor Labs)

You will, of course, have already read the latest article in our DSP series with much interest. There we have attempted to explain nearly all the ins and outs as comprehensibly as possible. Nearly all, that is. For some details there was no space in the article. However there are a few practical details that we do not wish to keep from you.

A few years ago, from the time of our famous Class-D Clarity amplifier, we, as designers, did not have much experience ourselves using SMD components in PCB designs. Because the Class-D amplifier operates with large currents and at high frequencies it was important that the PCB was as small and compact is possible. One method was to avoid separate vias by combining them with the pads of the resistors and capacitors. When using our product assembly service for placing the SMD components on the PCB, the relevant company pointed out the potential production problems this could cause. There is the risk of 'tomb-stoning', i.e. the components standing up on end during soldering. This is caused by the capillary action of one via, which will pull an SMD component upright. Fortunately we had none of these problems on this particular PCB. We have, however, avoided doing this since then.

In addition to tomb-stoning the DSP board has another issue that we would like to pay some attention to. We used an IC in a so-called HTSSOP package, which contains a Thermal Pad (the TLC5926 LED driver). This type of package has an 'exposed pad' on the bottom of the package. This enables the internal heat generated by the IC to be conducted to the outside, where this surface is usually also the ground connection of the IC. The intention therefore is for this pad to be soldered to the plane the PCB designer is supposed to pour under the IC. This carries the potential risk of the component 'floating away' when the solder liquefies during the reflow process. This risk is bigger if too much solder paste is applied. On the other hand, using insufficient paste is undesirable as well since it reduces the coupling with the (ground) plane and therefore worsens the heat transfer. The best way is to apply the solder paste with the aid of a stencil: a sheet of a certain thickness and perforations at those places that need to have solder applied. By filling all the perforations with solder and subsequently wiping the sheet we

apply exactly the correct quantity of solder paste.

In some cases it is even more difficult: for example, with our DSP board, there is the odd via in planes directly underneath ICs. This improves the heat transfer even more (additionally making it easier for the heat to be dissipated by the copper plane on the other side of the board). But there is the risk that the solder will flow away to the other side through these vias and not enough remaining on the exposed pad to ensure that the IC is properly connected to the PCB.

Placing a solder mask around the vias is one solution to ensure that the solder will not flow away through the via. Another potential way to avoiding this is to keep the vias as small as possible. The smaller the hole in the PCB, the harder it is for the solder paste to flow away through the via.

While on small vias: did you know that our PCB manufacturer did not allow us to make the vias smaller than 0.25 mm? This also requires a minimum copper annulus of 0.15 mm, so that the entire via is a little bigger than 0.5 mm in diameter. The big boys are probably laughing at this, but you have a look at the photo yourself where the scale will be clear...

(110551-I)

E-Blocks go Twitter
Using embedded wireless networks

By Ben Rowland (UK)

In this project we look at how you can easily link a wireless network card to your microcontroller system to develop a website containing useful information about the environment and even post messages on Twitter.

At our local sailing club there are more than 1000 members. One of their difficulties is that they do not know when the conditions are suitable for sailing. It's not just that there has to be wind for sailing, but health and safety regulations dictate that a qualified life guard (one of the members) be present when anyone is on the lake. To solve this problem we proposed that a website could be created to inform members whenever the lifeguard enters or leaves their post at the sailing club, along with local weather conditions and other sailing information. Also on the website is a link to the popular social networking site Twitter [1], so that one member can let others know that he/she is going to the club and that conditions are suitable. It was also proposed that the website should include a web camera, mounted to a web visitor controlled servo motor. Ideally, to provide visual information, the servo motor must also respond to control commands from the visitors of the website.

Hardware used

To get the project up and running on the bench I used a selection of E-blocks that

Figure 1. The prototype system based on E-blocks.

you can see in **Figure 1**. This consists of an EB006 Multiprogrammer fitted with a PIC18F4455, an EB003 Sensor board, an EB007 Switch board, an EB005 LCD board, an EB059 Servo interface board and an EB069 Wireless LAN board. For the sake of the prototype the temperature reading comes from a stainless steel temperature probe inserted into the sensor board.

The light and wind-speed readings come from the LDR and potentiometer on the sensor board. Finally, the lifeguard sensor is simply a switch (SW0) from the EB007 board. This switch could then eventually be placed under the lifeguard's seat to allow the Twitter messages to be sent out automatically with no user interaction.

The Matrix Multimedia Wireless LAN E-blocks board is a key part of the system: this allows easy access to the sailing club's wireless network and is fully supported by Flowcode V4. The E-block can be used to host a wireless network or join an existing wireless network. In the network host mode there is no simple way of allowing Internet access so for the purposes of this article we will be using the client mode.

Elektor Products & Services

- E-Block Multiprogrammer (EB006)
  (Note: PIC18F4455 not included)
- E-Block Sensor Board (EB003)
- E-Block Switch Board (EB007)
- E-Block LCD Board (EB005)
- E-Block Servo Interface Board (EB059)
- E-Block Wireless LAN Board (EB069)
- Flowcode for dsPIC/PIC24: #TEDSSI4
- Flowcode program file: 110388-11.zip (see [2])

Pricing and ordering details at www.elektor.com/e-blocks

Setting up the wireless LAN board

In this article we want to be able to communicate with the E-blocks system via the Internet, so to begin with we first need to connect the wireless E-block to the existing local wireless network as seen in **Figure 2**.

The WLAN board can act as a server, serving pages wirelessly to other wireless LAN devices, or as a client device communicating with a remote server. To begin this process you first configure the Wireless LAN component to be an end device. For most systems the Flowcode WLAN component property configuration shown in **Figure 3** will be correct.

To allow WLAN Internet requests, you

unsecured, then an empty null string can be used for the key. You can see this program in **Figure 5**.

Configuring your router

Once the system is up and running, you should be able to view web pages served by the embedded system on the local network. To see the pages on the local network you will first have to discover the IP address of the WLAN module, which should be shown in the DHCP client list on your router. Entering the IP address of the WLAN module into an Internet browser will reveal the WLAN configuration utility. This is similar to the configuration utility on a standard router and will allow you to check all of the Flow-
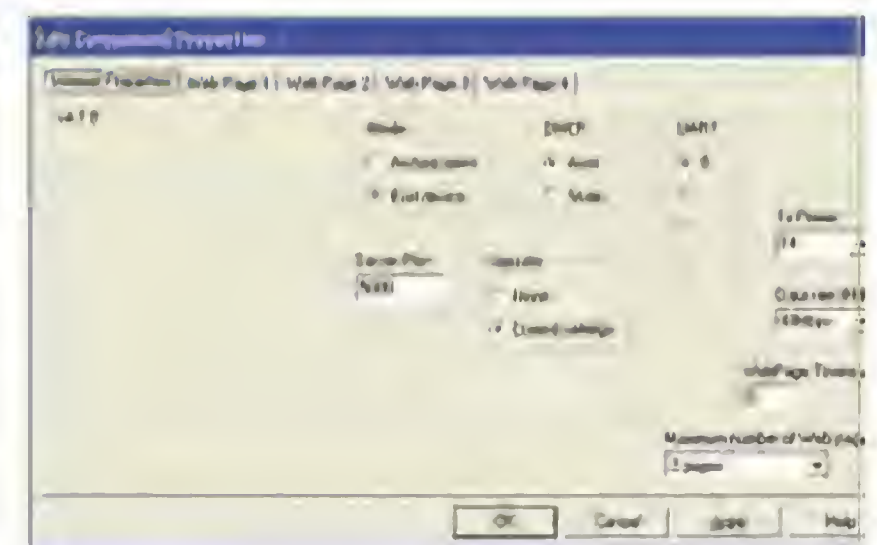
Figure 3. Wireless LAN Flowcode dialogue.

server port is 5000:

http://192.168.0.4:5000/

Connecting to the Internet

Once we have confirmed that the WLAN module is serving pages correctly we can then configure the router to allow the module to be addressed via the Internet. Doing this means you can access the embedded system from anywhere in the world. To aid in configuring your specific router there is a website at http://portforward.com that guides you through the steps you need to
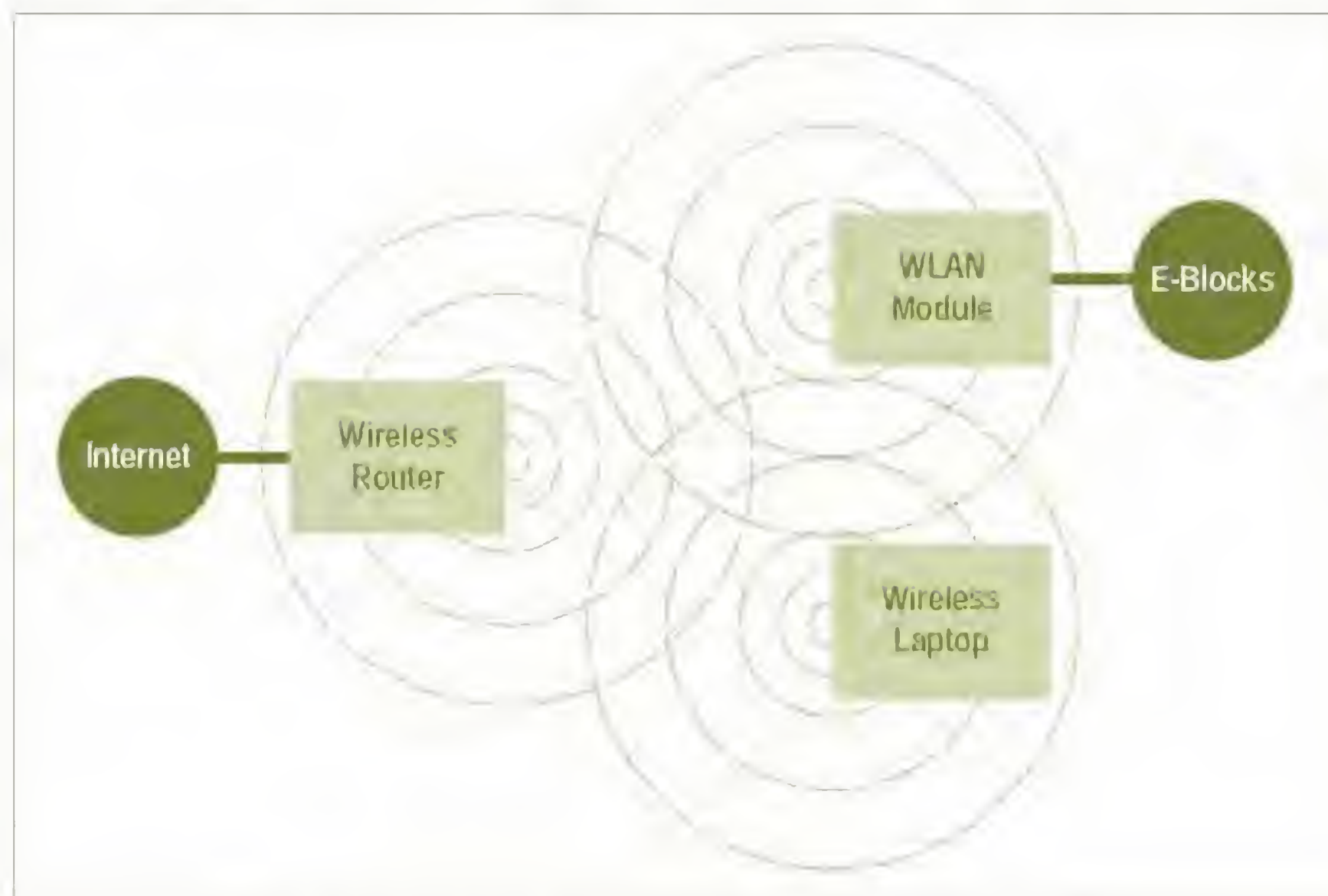
Figure 2. Local area network (LAN) wireless configuration.

have to initialise the WLAN module, which involves resetting the device and passing the set of Flowcode initialisation properties to it. You can see the routine for this in **Figure 4**. Once this has been done, the Connect_To_SSID component macro is used to connect the module to a host wireless router. The connect macro requires two parameters, the first being the network name (SSID) and the second being the wireless security key. If the network is

code settings have been loaded into the module correctly.

To see actual data pages served by the system you need to manually add the specified server port to your browser URL. Here is an example URL address where the WLAN module's IP address is 192.168.0.4 and the

Figure 6. Wireless LAN Flowcode HTML page.
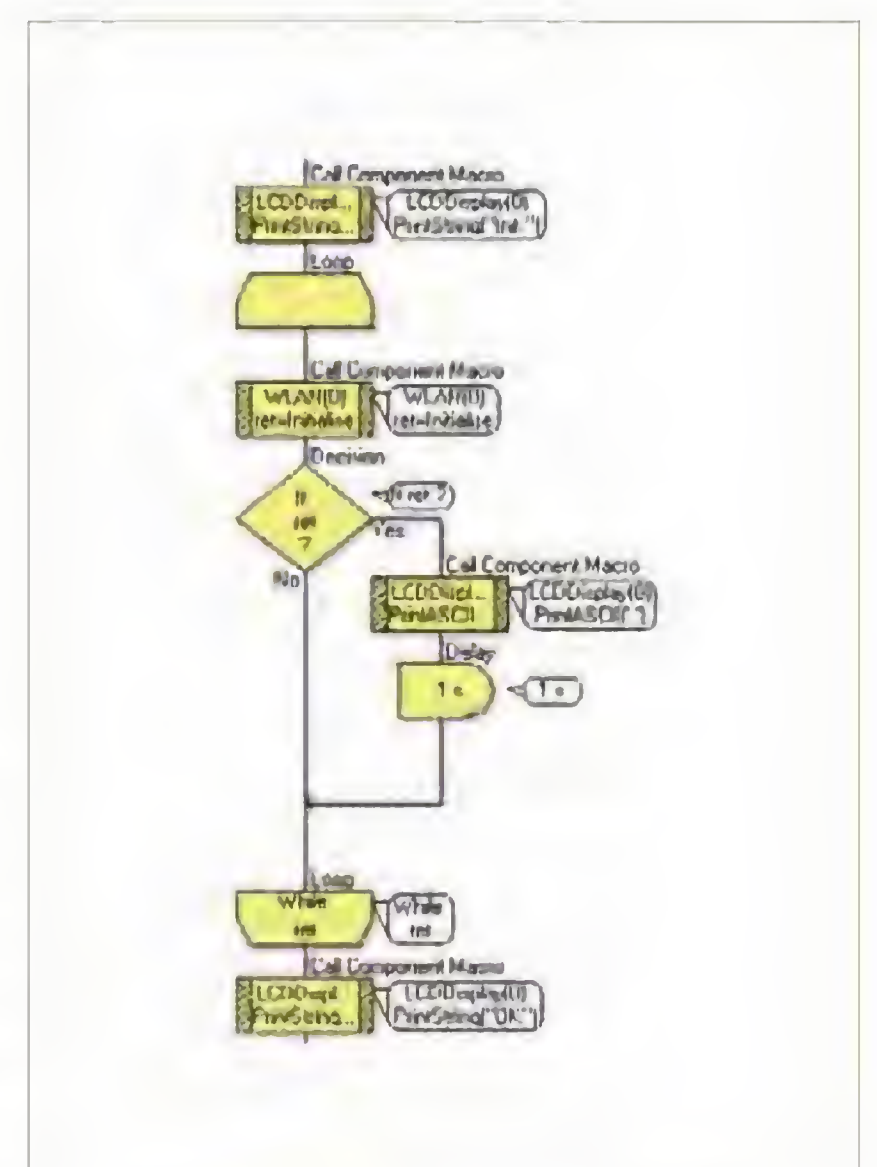
Figure 4. WLAN Initialising in Flowcode.

perform to allow web access to the embedded system. They also offer a paid service to help you getting up and running. Your router manual will also contain a good source of information on how this is done for your hardware.

To connect to the embedded system via the Internet, you must enter the URL of your local Internet connection as detailed by your router. As the IP address supplied by your Internet service provider can change regularly, there are free services such as http://no-ip.com that will provide you a free static domain name that will automatically forward you to your current IP address. The WLAN module directly supports this kind of functionality named Dynamic Domain Name System (DDNS) so you can enter your no-ip username and password into the module's configuration utility and this will automatically keep your IP address synchronised to your domain name.

Setting up the web pages in Flowcode

The web page content is configured by entering HTML and Javascript code directly into the Flowcode WLAN component. You can see an example of this in **Figure 6**. The variables used in the web page like temperature and wind speed link directly to Flowcode program variables. Outgoing variables are controlled using a Flowcode component macro and inserted into the HTML using a percentage character '%' followed by an index number. E.g.,

```
Temperature = %0.
```

On the other hand, incoming variables are controlled by adding the variable's index and value to the URL — similar to how variables are passed in PHP. E.g.,

```
index.htm?0=255&1=39
```

Page requests are serviced by regularly calling the Check_For_Page_Requests component macro within the Flowcode program. You now have a versatile microcontroller system that can communicate over wireless with local networks and the Internet alike, and you're able to pass values in and

Figure 7. Sailing Club Weather Report.
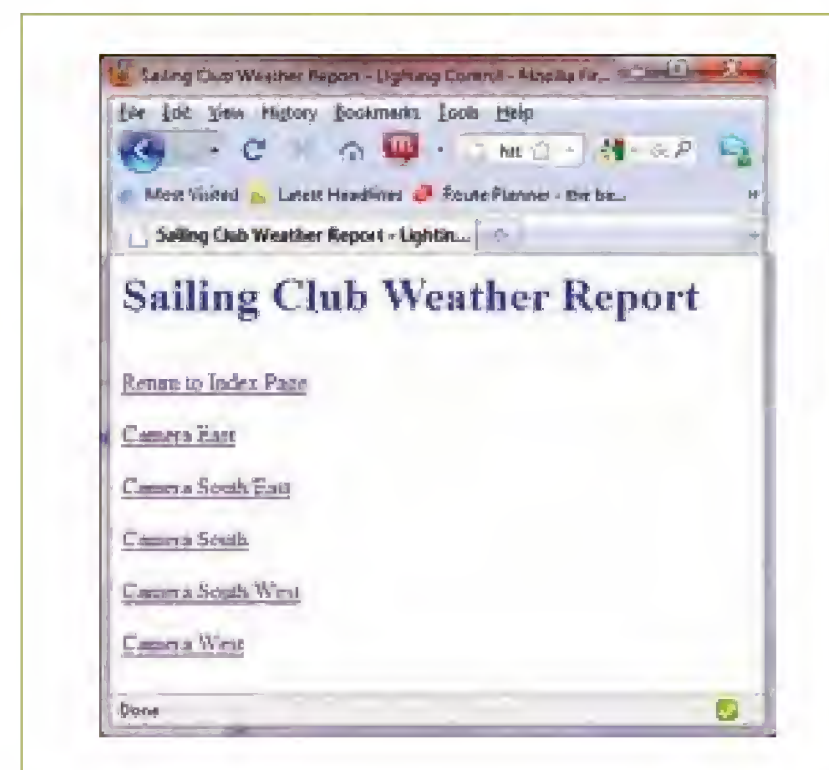
Figure 8. Sailing Club Camera Control.

out of the system. Examples of the web pages served from the microcontroller can be seen in **Figures 7** and **8**. The main page


Figure 5. WLAN Connect to SSID.

shows the weather information and a link to a sub page which allows users to control the

direction of the camera. The main page also shows a Twitter link which can be used by one member to send an 'attendance' message to all other Twitter feed subscribers.

Creating a Twitter link

Next, to create the Twitter post detailing what is happening in the system for all feed members. This was done by creating a Twitter button on the web page and populating this with the data collected by the sensors. Then when users visit the website and click the Twitter button, they can send out a message to any of their followers detailing the conditions at the club. I did try to get the system to automatically send out Twitter messages whenever the lifeguard entered or left the club, but I could not get this to work reliably so it was dropped for the time being.

Conclusion

All this kit is now up and running on the bench and neatly communicating to the web. The next step is to get hold of an anemometer and to take the hardware into the field...

The Flowcode programs are — as always — available from the Elektor website [2].

(110388)

Internet Links & Literature

[1] www.twitter.com

[2] www.elektor.com/110388

     http://portforward.com

     http://no-ip.com

Audio DSP Course (3)
Part 3: DSP board

By Alexander Potchinkov (Germany)

In this instalment we present the DSP board. It is the platform for the applications that will be described in later instalments of this series, and it is also intended to enable you to develop your own initial digital audio signal processing applications, which we hope will be followed by many more. The DSP board can be used stand-alone as is, and even though it is an ideal learning platform, with its 24-bit signal processing capability for sampling rates up to 192 kHz and its high-performance interfaces, it is also suitable for applications with very stringent quality requirements for both signal to noise ratio and DSP computing power.

The DSP board is intended to be used for processing audio signals with a digital signal processor. The signals to be processed may be analogue, digital, or a combination of the two.

Figure 1 shows the block diagram of the circuit. The selected components allow the hardware to be limited to 13 ICs on a PCB measuring 97 by 66 mm, despite the impressive performance capability. Two-channel ADC and DAC ICs with 24-bit resolution and sampling rates up to 192 kHz are provided for processing analogue signals. These converters were selected on the basis of price, the lowest possible peripheral component count, and availa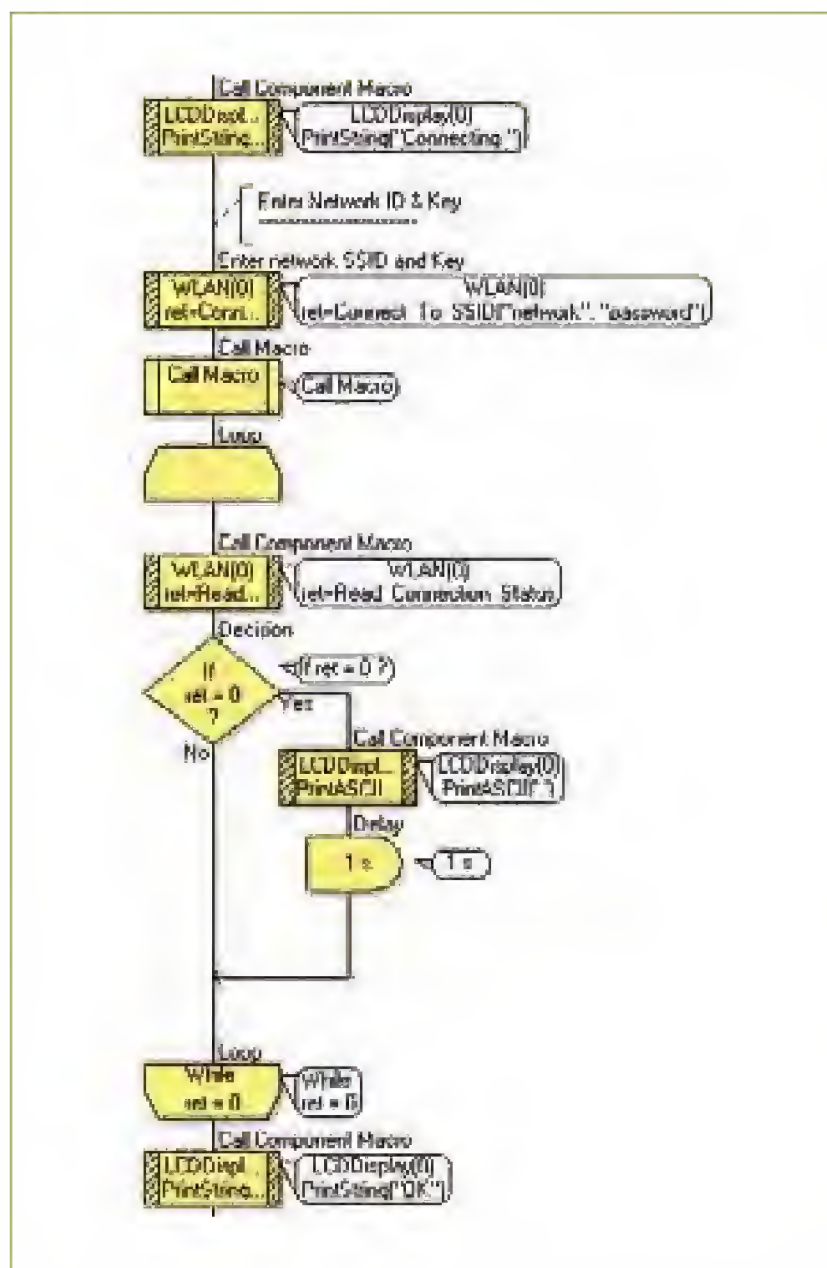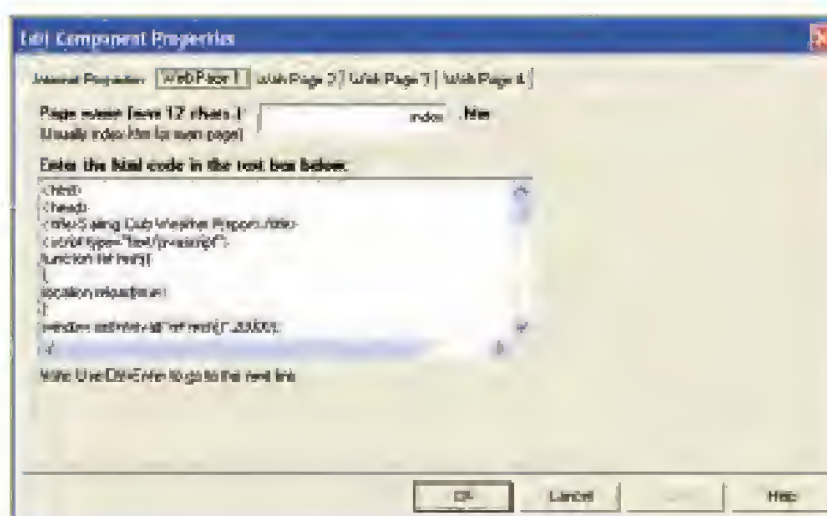bility. They are suitable for operation under hardware control and do not require different configurations for different sampling rates.

Figure 2 shows the audio signal paths supported by the DSP board. The DSP is the audio master, and on the input side it receives digital audio signals and analogue audio signals converted into I²S format. On the output side it provides signals for simultaneous conversion into analogue and digital audio signals. If you consider only the audio signal paths and ignore the signal processing functions, the DSP can be imagined to act as a three-position source selection

switch. In position 1 it supplies analogue audio signals to the audio outputs, in position 2 it supplies digital audio signals, and in position 3 it supplies signals generated by the DSP itself.

Signal processing and signal transmission both require the use of the DSP.

Digital signals can be input and output using optical or electrical ports at the user's choice. On the input side these two modes (optical and electrical) are mutually exclusive, while on the output side both modes can be used in parallel. An asynchronous sample rate converter (SRC) on the input side converts digital signals having a wide range of sampling rates into digital signals that are sampled at the rate used for digital signal processing. The 'professional quality' sampling rate of 48 kHz is used for applications described in this series of articles because it allows sufficient bandwidth along with high computing power.

Signal processing is performed by a Freescale DSP56347, which is specifically designed for audio signal processing and can be programmed to perform any desired function. If you wish to perform digital signal processing at a different sampling rate,

all that is necessary is to change the settings of two interface configuration registers (e.g. for 96 kHz sampling, only one bit in each register needs to be changed).

Many different applications are possible with the DSP board. For example, you can connect a CD player directly to the board and use it with a LED board described in this series of articles to construct a VU meter; you can connect a digital microphone and an amplifier (with speakers) to the board and use it to suppress feedback howl; or you can use the board to compute the harmonic distortion level of an analogue signal and show the result on a display.

Despite the manifold application possibilities of the DSP board, it needs only a relatively small number of components — thanks to the advanced state of development of modern digital audio IC technology. The block diagram shows four signal processing blocks: audio signal input and output, digital signal input and output, the DSP block, and the DSP peripherals. These blocks are described below to give you an understanding of what is on the DSP board and what can be implemented using the board.

Communication on the DSP board

The block diagram in **Figure 3** depicts the communication structures on the DSP board. Although this figure may at first glance appear surprisingly complex in light of the small size of the board. it provides a good indication of the versatility and application diversity of the design. There are two communication channels: an I²S audio bus for audio data, and an SPI bus for control and miscellaneous data. The audio bus has five nodes, with the DSP acting as bus master and the other four nodes acting as slaves. The audio bus clock lines are shown in black and divided into two groups, although clock lines of the same type can be joined together on the DSP board. This is indicated symbolically on the drawing by labels in parentheses and dashed connecting lines. We split the clock signals into the clock signals used on the board and the clock signals connected to I²S port K6 to allow this port to also be used for GPIO if an audio port is not needed. If the port is used as an I²S port, the following clock lines can be interconnected on the DSP board: HCKR to HCKT, FSR to FST, and SCKR to SCKT.

The upper six line of the audio bus lines the audio data lines. Three of them are used internally on the board, while the other three are connected to port K6. The connections to the analogue and digital audio interfaces of the board are shown below the I²S bus in the figure.

The second bus is the SPI bus, which has four nodes. Here again the DSP is the bus master and provides the bit clock for shift register operations. The three slave nodes are the SEEPROM (which acts as rewritable nonvolatile memory with low access speed), the sample rate converter, and SPI port K7, to which any desired external SPI slave device can be connected. In this course we use this port to supply data to a LED bargraph display.

Slave nodes on an SPI bus must be enabled or disabled by chip select signals to prevent
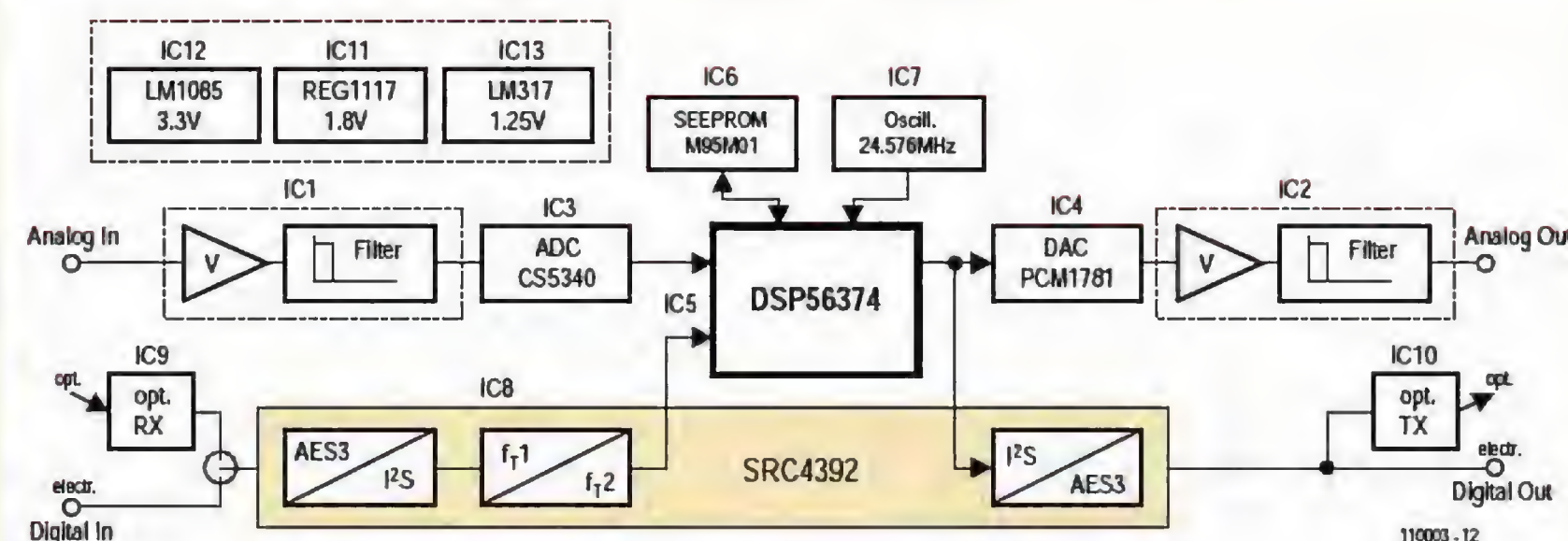
Figure 1. Circuit block diagram. The signal interfaces support two-channel audio.
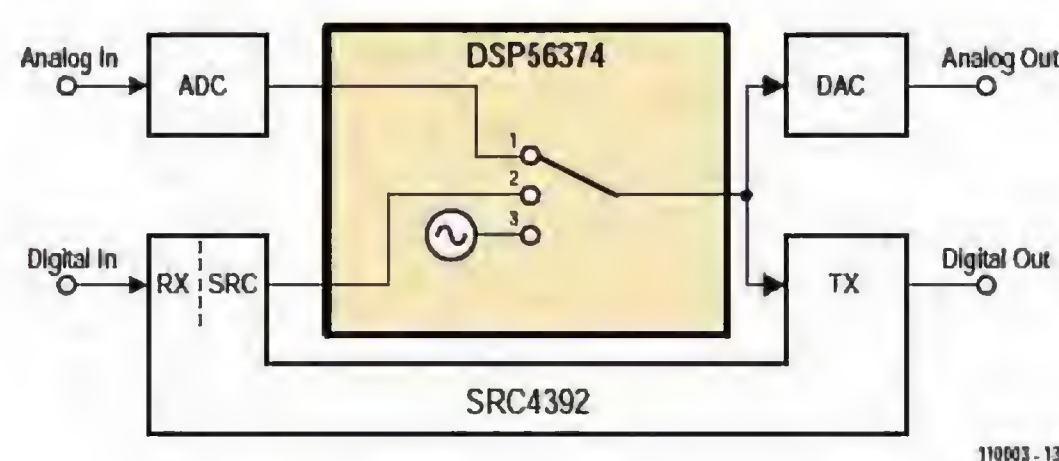
Figure 2. Audio signal paths.

them from concurrently driving the MISO line. These chip select signals are generated by the DSP. Additional lines are used for SRC control and handshaking. One of these lines is used to reset the SRC, which for example must be done before it is configured. The SRC uses the other line to indicate to the DSP that a digital audio signal is present on its digital audio input.

Analogue signal inputs and outputs (IC1–IC4)

The two-channel analogue signal input stage with pin headers K1 and K2 is built around operational amplifiers IC1a and IC1b and ADC IC3 (type CS5430); see **Figure 4**. The analogue portion of the ADC is powered from the 5 V supply rail. The two operational amplifiers are wired for unity gain and add a DC offset equal to half the supply voltage to the input signals after AC coupling capacitors C1 and C2. They also serve as low-impedance signal sources for the ADC and the analogue anti-aliasing sub-filter of the oversampling ADC.

The ADC operates with fixed settings as an audio slave device, which means that the DSP provides the necessary audio clocks, consisting of the master clock, the bit clock and the left/right (LR) clock, which corresponds to the sampling rate. The ADC is operated in one of three modes depending on the desired sampling rate: single-speed, double-speed or quad-speed. For each of these modes, specific ratios between the master and LR clock frequencies can be
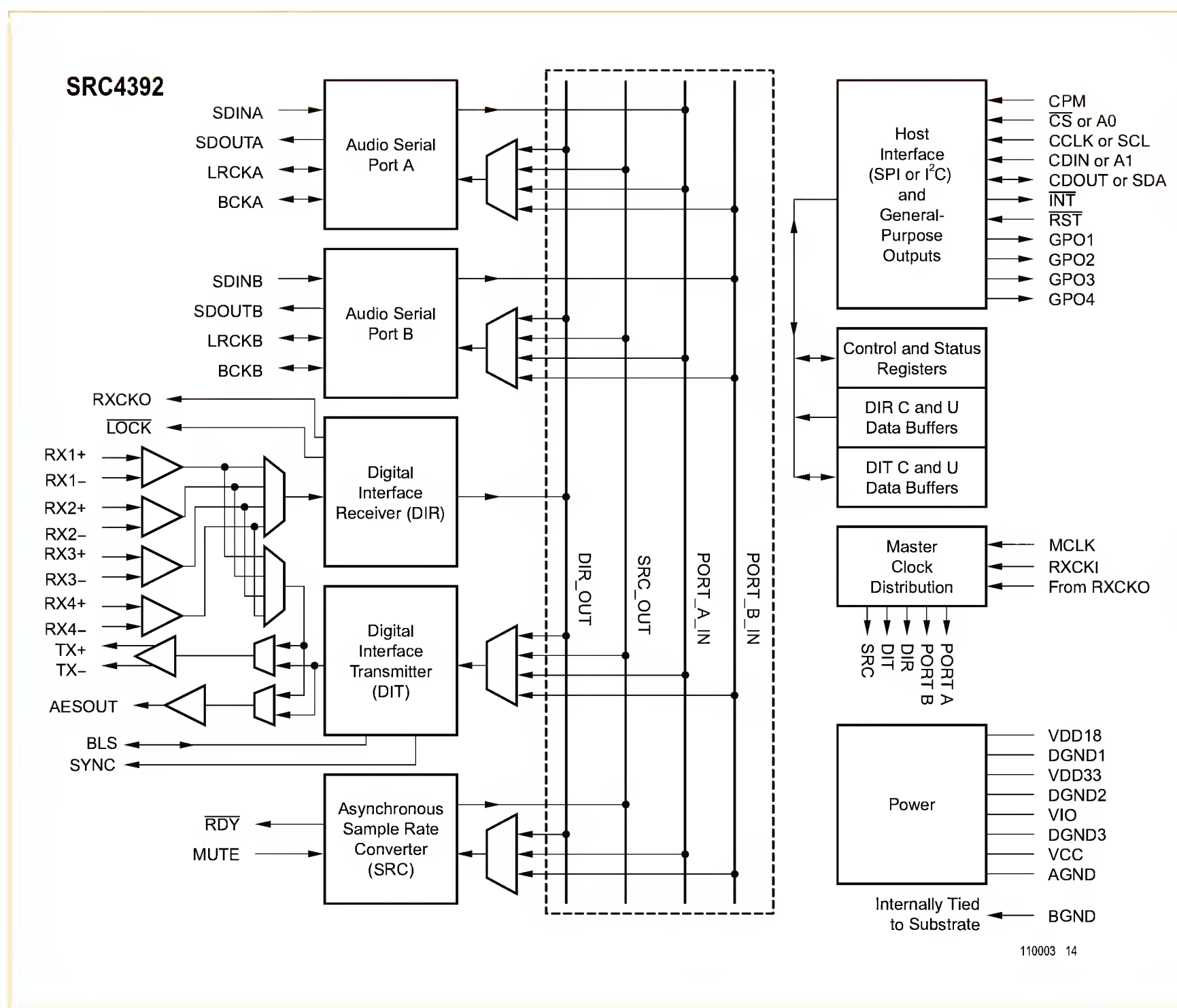
Figure 3. Communication on the DSP board.

specified over a range of 64 to 512.

The ADC is able to automatically detect the ratio of the clock signals generated by the DSP, which is determined by the DSP firmware. According to the data sheet, the ADC has a dynamic range of 101 dB and a THD + N level of −94 dB, which is sufficient even for applications with stringent requirements. The peak-to-peak signal level with a sinusoidal signal is 0.53 to 0.59 times the supply voltage, with an average level of 2.8 V.

On the digital side of the ADC, which is powered from the 3.3 V supply rail for digital peripheral devices, I²S audio mode is selected by pull-up resistor R14 connected to pin 4. I²S audio mode is used for all audio signals on the DSP board. Power-up reset is provided by network R13/C20.

The two-channel audio signal output stage with pin headers K3 and K4 is built around DAC IC4 (type PCM1781) and operational amplifiers IC2a and IC2b. With an audio word width of 24 bits, the DAC has a dynamic range of 106 dB and a typical THD + N level of 0.002%, equivalent to approximately -94 dB. It can be operated at sampling rates ranging from 5 kHz to 200 kHz. The DAC is configured for I²S and is controlled by the DSP as an audio slave.

The audio clocks generated by the DSP are the same for all audio interfaces (ADC, DAC and SRC). The four configuration pins (1–4) are configured for I²S, de-emphasis off, and mute off. The DAC can automatically detect the ratio of the master and LR clock signals generated by the DSP, which allows it to be used without separate configuration.

The reconstruction filters (low-pass filters used to convert the oversampled digital signals into analogue signals) are implemented using the two operational amplifiers. They are second-order Butterworth filters with a DC gain (A_0) of 1. The stop frequency is approximately 30 kHz with the specified component values. The stop frequency has intentionally been set relatively low because DACs of this sort generate predominately high-frequency noise, so the bandwidth should be kept as small as possible.

Changing the filter characteristics is not difficult. If you wish to maintain the Butterworth characteristic, the Q (quality) factor is 0.7071 ($1/\sqrt{2}$). Start by specifying the DC gain, the value of capacitors C26 (for the left channel filter) and C27 (for the right channel), and the stop frequency f_0 or ω_0 (= $2\pi f_0$). The values of the other components can then be calculated (taking the left chan-

nel as an example) using the formulas

$$C28 = C26 / (4Q^2 \times (1 + A_0)),$$
$$R24 = 2Q / (\omega_0 \times C26),$$
$$R25 = (1 + A_0) \times R24,$$
$$R23 = (1 + A_0) \times R24 / A_0.$$

If $Q = 0.7071$ and $A_0 = 1$, these formulas simplify to

$$C28 = C26/4,$$
$$R24 = 1.4142 / (\omega_0 \cdot C26),$$
$$R23 = R25 = 2 \times R24.$$

The component values for the right channel filter can be calculated in the same way. The DC offset for the filters is taken from pin 13 of the DAC.

Particular attention has been given to protecting the analogue outputs from damage if they are connected to microphone inputs with a phantom supply voltage of up to 48 V. To provide sufficient tolerance in both directions against DC voltages up to these levels, the output capacitors have a suitably high rated voltage and are connected in reverse series. The dual Schottky diodes D1 and D2 protect the operational amplifiers against the effects of an output cable short in such a situation, which would otherwise cause the capacitors (charged to around 48 V) to discharge through the operational amplifiers.

At the full-scale DAC output level, the output voltage level with a sinusoidal signal is approximately 3.9 V peak to peak.

Digital signal inputs and outputs (IC8–IC10)

Audio data is transported internally on the DSP board in I²S format. The I²S bus is a synchronous serial bus with three bus lines: LR clock, bit clock, and audio data. A different format called Digital Audio must be used for signals entering or leaving the board. It is designed to allow data to be transmitted using a single fibre link. This requires two converters, which convert Digital Audio signals to I²S signals or the other way around. These two converters, which are know as the receiver (RX) and the transmitter (TX) below, are housed in the SRC4392 (IC8). Before describing the operation and control of IC8, we should first briefly describe the digital audio interfaces of the DSP board. An

optical receiver (IC9) and an optical transmitter (IC10) are located on the board. They can be connected to the RX and TX stages of IC8. The board also has ports for electrical signals. Either the optical input or the electrical input can be selected by a jumper on pin header JP1. In order to use the optical or coaxial input, a jumper must be fitted between pins 1 and 2 of pin header K9 (connecting the minus input terminal to ground) so that the RX1 input of the SRC is tied to ground via C54. This jumper should be removed if a balanced input signal is used.

Either an unbalanced signal or a balanced signal can be connected to pin header K9, which has a 75 Ω termination (R54). To comply with the relevant standard, a balanced signal should be input using a standard pulse transformer and two 18-Ω resistors wired directly to the XLR input connector. You can make your own pulse transformer from a ferrite ring core and a few centimetres of enamelled copper wire. A differential RS422 signal output is available on pin header K10. It can be used together with a pulse transformer and a 110-Ω resistor to provide an AES-3 balanced output, or with a resistor and a 10-nF capacitor to provide an unbalanced S/PDIF output. Additional I²S ports that can be controlled by the DSP are also brought out to pin header K6. They can be used for connection to standard ICs with an I²S interface.

IC8 is a very high-performance interface IC described by its manufacturer as a 'two-channel asynchronous sample rate converter with integrated digital audio interface receiver and transmitter'. Its block diagram (see **Figure 5**) shows several function blocks and four audio data busses. The blocks for digital audio input are on the left side.

The two Audio Serial Ports (A and B), of which only one (port A) is used on the DSP board, provide the links to the DSP. Port A is operated in I²S mode and provides the audio input and output paths for the DSP. Two signal paths for digital audio signal input and output are available in the SRC (IC8) on the DSP board. The first path runs from the differential digital inputs pins (RX1+ and RX1–) of the Digital Interface Receiver (DIR) over the DIR_OUT bus to the Asynchronous Sample Rate Converter (SRC), and from there
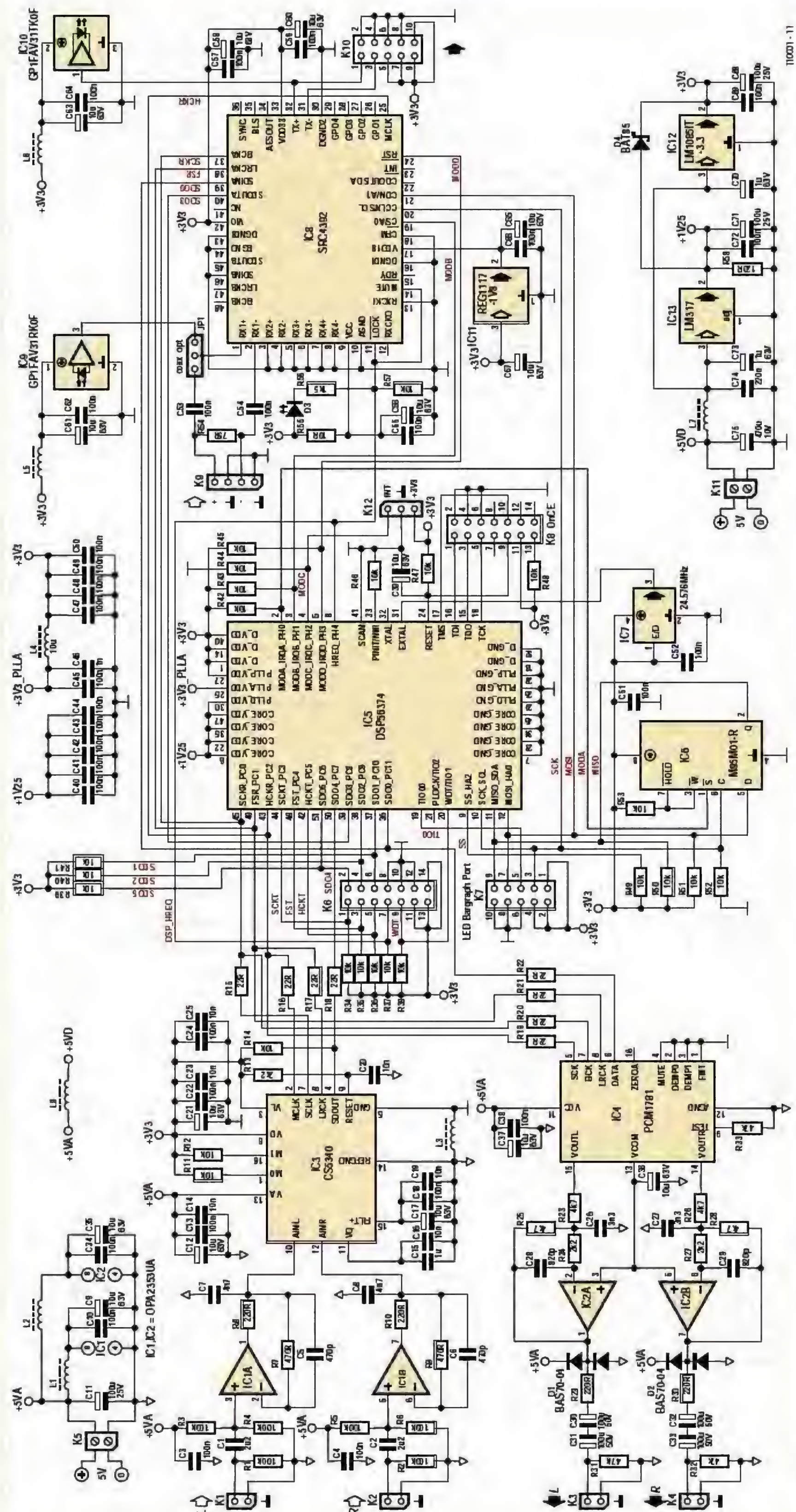
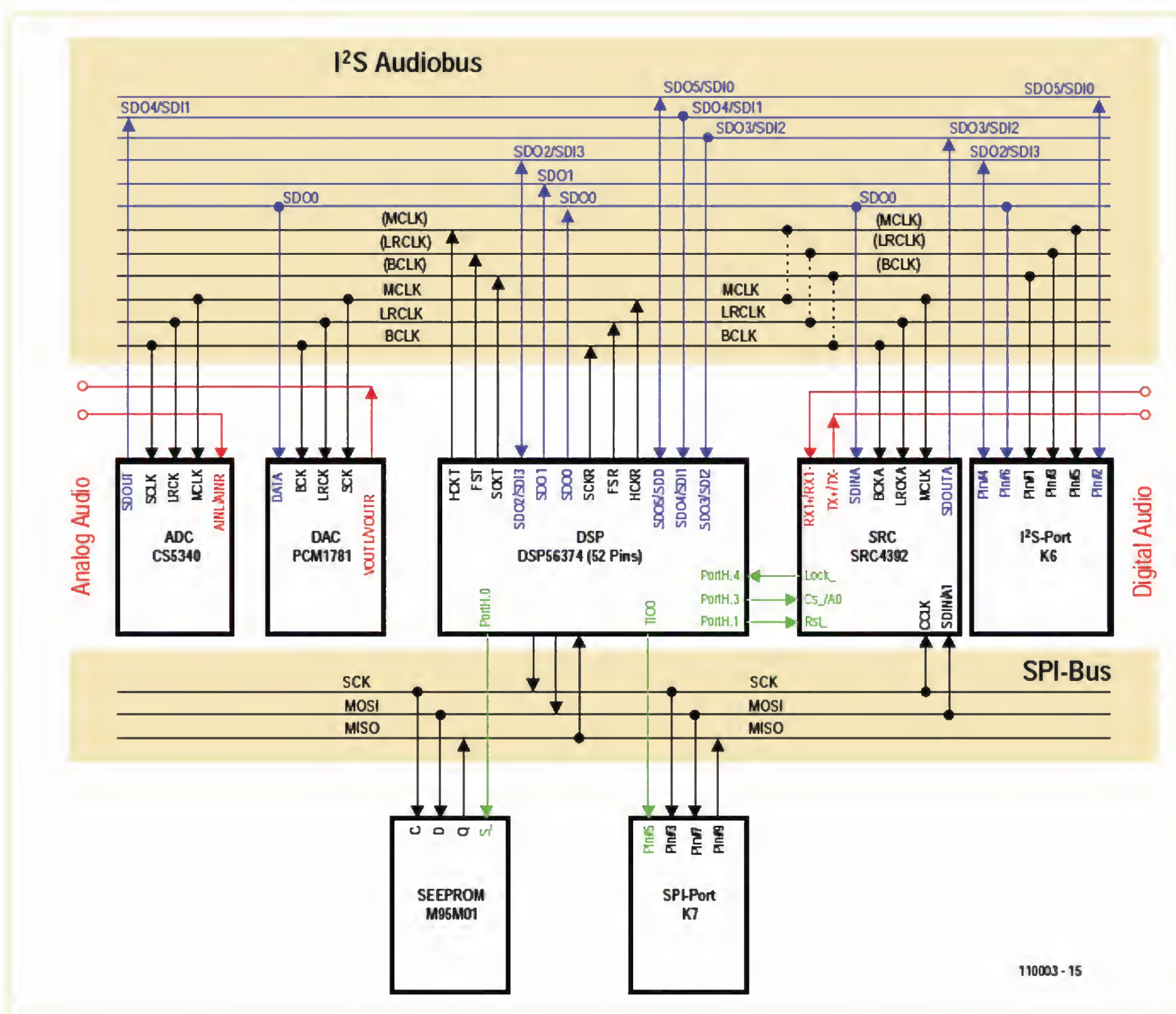Figure 4. Complete circuit diagram of the DSP board.

Figure 5. Block diagram of the sample rate converter.

over the SRC_OUT bus to Audio Serial Port A and its data output pin SDOUTA, which is connected to the DSP.

Audio data from the DSP follows the second path, which starts with audio data input into Port A. From there it is transported over the PORT_A_IN bus to the Digital Interface Transmitter (DIT), where it is converted to Digital Audio format for output from the board.

The DIR converts audio signals in Digital Audio format to audio signals in an internal format that can be read by the SRC, and it synchronises the input port of the SRC. The output port of the SRC is synchronised by the DSP, which sends audio clock signals Port A of IC8.

The Asynchronous SRC interpolates the audio signals. This is done by generating a quasi-continuous signal, similar to an analogue signal, from the incoming audio signal and sampling this signal at a different sampling rate to form the output signal. Interpolation can be performed as a purely digital process because it does not require an analogue signal.

The SRC can process audio signals with sampling rates ranging from 20 kHz to 216 kHz and convert them to a different sampling rate, such as 48 kHz or 96 kHz. These rates

are commonly used in professional applications, and they are used by the DSP for audio signal processing. The DSP is supplied with audio data from Port A, and it returns audio data to the SPDINA pin on the same port.

If you consider the functional scope and audio quality (24 bits, equivalent to 140 dB) of the selected SRC component, you can see that it belongs to the top end of the performance range of comparable ICs. As you might imagine, there's also a downside to this: a device with this degree of complexity cannot operate under hardware control; it requires configuration under software control. This is done by writing 52 bytes to the control register bank, which is done using the SPI protocol. The relevant registers and their functions are shown in table 1.

The remaining registers are filled with zeros. Two $01 bytes must be sent before actual register configuration starts with register $01, so a total of 54 bytes must be transmitted. The byte sequence is stored in the file src4392.tab.

The Lock signal on pin 11 of IC8 is important because it indicates whether a valid digital audio signal is present at the receiver input. This signal is used in the DSP firmware to determine whether to select digital input or

analogue input. If a valid digital input signal is present, the SRC output signal is processed; otherwise the ADC output signal is processed. The presence of a valid digital input signal is indicated by LED D3.

IC8 operates from a separate 1.8 V supply voltage generated by low-dropout voltage regulator IC11 from the 3.3 V supply voltage.

DSP and clock oscillator (IC5 and IC7)

The audio DSP, a Freescale DSP56374, is a highly integrated IC requiring only a few peripheral devices. The DSP clock is derived from the 24.576 MHz clock generated by crystal oscillator IC7, which is multiplied by a factor of 6 in the PLL frequency multiplier circuit integrated in the DSP to produce a 147.456 MHz clock signal. At an audio signal sampling rate of 48 kHz, 3072 instruction clock cycles are therefore available to the user in each sampling interval for signal processing.

The audio clocks — master clock, bit clock and LR clock — used by the ADC, DAC and SRC are derived from the processor clock signal by dividers in the DSP. The DSP acts as audio master, which among other things allows the sampling rate to be set to 48 kHz, 96 kHz or 192 kHz. Incidentally, the latter rate has little technical relevance and tends to be used by marketing strategists to promote ADC and DAC devices that do not support 24-bit audio resolution and therefore try to convince potential users by boasting especially high (but unnecessary) maximum sampling rates.

The schematic symbol for the DSP in Figure 4 has the pins arranged in various groups according to their function. The pins for the supply voltages and ground connections are shown at the top and bottom. The DSP uses a supply voltage of 3.3 V for its peripheral circuitry and a lower supply voltage of 1.25 V for the processor core. This allows the power consumption to be kept relatively low, even at a high processor clock frequency.

A group of 12 pins for audio interface functions is located on the left side. Half of them are used for the data lines of six I²S ports

Table 1. Programming the functions of the SRC.			
Address	**Content**	**Block**	**Function**
$01	$37	Alle	Enabling Port A, TX, RX and SRC
$03	$31	PortA	Format I²S, Source=SRC, Slave, Mute off
$07	$60	TX	Source=PortA, ClockRatio=512, ClockSource=MCLK
$09	$03	TX	BufferSource=SPI
$0D	$08	RX	Input RX1, reference clock=MCLK
$0E	$10	RX	Clock free runs after loss of clock
$0F	$22	RX	Reference clock=24.576MHz
$1B	$07	GPO1	Receiver non-valid data active high
$2D	$42	SRC	Source = DIR, reference clock = MCLK, mute off, tracking on
$2E	$03	SRC	64 samples group delay, true decimation, de-emphasis off

that serve as inputs and outputs, of which at most four ports can be used as inputs. The other half provide the audio clock signals, consisting of two sets of three clock pins. It is possible to clock the inputs and outputs separately, which among other things allows users to work with two different sampling rates. We do not make use of this possibility; the same clock signals are used for the inputs and the outputs on the DSP board. Accordingly, the SCKR, FSR and HCKR outputs provide the I²S bit clock, LR clock and master clock signals. The DSP software can be used to set the clock signals as desired by configuring the peripheral device registers appropriately.

The I²S signal on SDO0 (pin 36) is used on the DSP board to drive the DAC and the SRC TX stage, which converts audio data to Digital Audio format in the SRC. Pin SDO4, which is configured as an input, connects the output of the ADC to the DSP, while pin SDO3 connects the output of SRC RX stage in IC8 to the DSP. This gives the DSP access to both of the board's signal inputs (analogue and digital). The three clock lines are used by IC3, IC4 and IC8, so the ADC, DAC and SRC operate synchronously. Pins TIO0 and WDT/TIO1 in the adjacent set of pins are fed out to pin headers and can be used for the DSP timer system, for a watchdog timer, or as GPIO.
The set of pins from SS_HA2 to MOSI_HA0 forms a synchronous serial port, which is used on the board as a bidirectional SPI

interface for communication with various peripheral devices. This includes writing configuration data to IC8, which is selected by the signal on the MODB pin of the DSP. The serial EEPROM is also connected to the SPI bus and can be read and written. It is selected by the signal on the MODA pin of the DSP. Finally, in one of the projects described in this course we use the SPI bus to write data to a LED board with two 40-LED bargraph displays.

Naturally, other SPI peripheral devices can also be connected to SPI pin header K7, such as a microcontroller with user interface components and a display, which could be used to enter configuration settings in the DSP program and show them on the display. The set of pins at the top right consists of MODA_IRQA to MODD_IRQD. After the DSP is reset, the processor reads the voltage levels on these pins and uses this information to select the boot mode. The combination of pull-up resistors R42, R43 and R45 and pull-down resistor R44 selects

booting from the on-board SEEPROM over the SPI bus. After the DSP has been booted, these pins can be used for hardware interrupts or GPIO. Three of them are used on the board for GPIO. The HREQ pin is connected to the Lock output of IC8, which indicates whether a valid audio signal is present on the digital audio input. The MODD pin can be used reset IC8, which must be done before it is configured. DSP terminal MODC_IRCQ is available on connector K12, allowing a method for hardware interrupts to be created. Howevber, if this is used, be sure to avoid conflicts with booting via the SEEPROM in bootstrap mode 11. This mode requires the logic level at this pin to be Low briefly after a reset, which is implemented with the aid of resistor R44 on the DSP board.
The signals on pins 31, 32 and 33 are not relevant to signal processing. They provide a connection point for the clock generator and determine whether the PLL clock multiplier in the DSP is active after the DSP is reset. The final set of pins (15 to 18) forms the
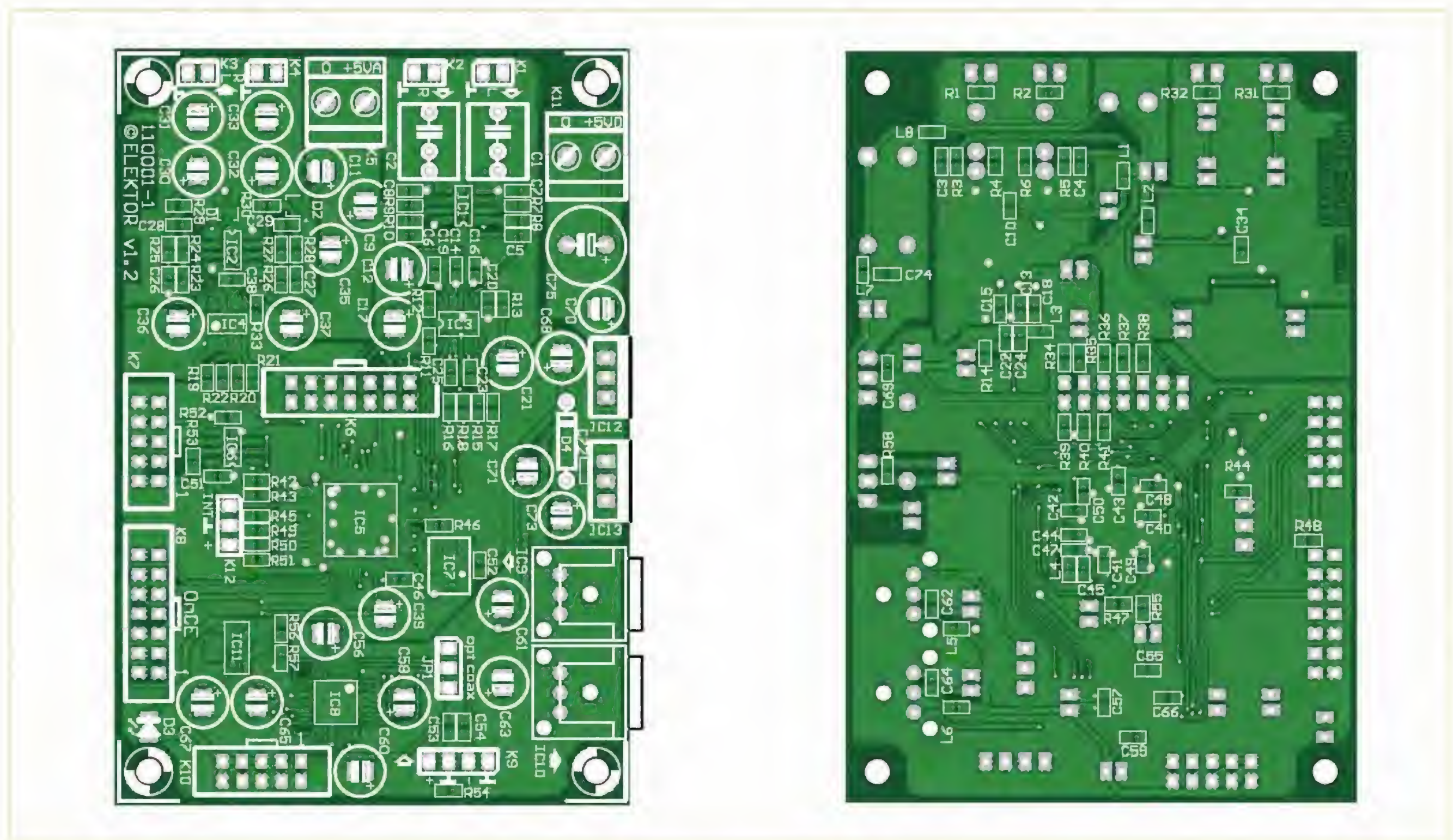
Figure 6. Component layout of both sides of the DSP board (see www.elektor.com/110003 for the components list).

debug port, which is fed out to pin header K8. It is used for communication with a debug program running on a PC, which can be used to download program code, to read all of the DSP registers, and to read and write the DSP's internal volatile memory.

DSP peripheral devices (IC6 and IC7)

The DSP peripheral devices can be described in just a few words because they consist of only two ICs: the previously mentioned clock oscillator (IC7) and IC6, a type M95M01 serial SPI 1-Mbit EEPROM.

The DSP has three banks of 6-Kword RAM with a total capacity of 442,368 bits ($3 \times 6,144 \times 24$), which provide less than half the storage capacity of the SEEPROM. Most applications do not require writing the entire DSP RAM with data from the SEEPROM when the board boots up. Furthermore, the SEEPROM can be read and written while the DSP is running. Although this is very slow compared to the DSP clock rate, it allows specific settings or the like to be read and written using data transmission over the SPI bus between a microcontroller-based user interface and the DSP board. However, writing data to the SEEPROM using the special autoincrement addressing mode requires some programming effort due to the paged structure of the SEEPROM, since the page structure must be taken into account when large volumes of data are read or written.

Power supply (IC11, IC12, IC13)

Finally, a few words about the power supply for the DSP board. The board has two separate 5-V power connectors: one for the analogue supply voltage (connector K5) and the other for the digital supply voltage (connector K11). These two connectors are linked by inductor L8, so only connector

K11 of the board supplied by Elektor needs to be connected to an external 5 V power supply. It may be possible to obtain a better signal to noise ratio for analogue signals by using separate power supplies, although this depends primarily on the quality of the power supply (or supplies). The other three supply voltages needed for the digital components — 3.3 V for the digital circuits, 1.5 V for the DSP core and 1.8 V for IC8 — are generated by linear voltage regulators IC11, IC12 and IC13.

The DSP board

As already mentioned, the DSP board is available from Elektor fully assembled and tested. Next month we will take our first steps on the way to putting the board to good use and describe a number of test routines. We will also say more about the necessary PC software and how to use it.

(110003-I)

FT232R USB/
Serial Bridge/BOB

By Antoine Authier (Elektor Labs)

You'll be surprised first and foremost by the size of this USB/serial converter — no larger than the moulded plug on a USB cable!
And you're also bound to appreciate that fact that it's practical, quick to implement, reusable, and multi-platform (Windows, Linux, etc.) — and yet for all that, not too expensive.

I don't think much of the various commercially-available FT232R-based modules. Too expensive, too bulky, badly designed, ... That's why I set myself the challenge to design this miniature in the form of a breakout board (BOB). One meaning of breakout is to escape, and in some ways, this board enables all the normally inaccessible signals within complex circuitry to 'escape' to the outside world so they can be accessed. Here, the complex circuitry is the legendary FT232R. The very one encapsulated within the plastic of the USB/TTL cables from FTDI, see [1][2].

The circuit diagram is based on the information in the FT233R data sheet [5] and makes it possible to produce all the applications described by FTDI (RS-232, RS-485, etc.), along with other ideas (JTAG, BitBang, etc.) which I may be suggesting to you in a later article.

110533 - 11

EagleCAD library

This library is useful in that it makes it easier for you to incorporate the bridge within your own projects. It's easy to use:

1. installation

- create the sub-directories "library/Elektor" in the installation directory "eagle" (subsequently called $EAGLEDIR).
- unzip the downloaded file into this new directory.
- In Eagle's "Options / Directories" menu, add ":$EAGLEDIR/library/ELEKTOR" to the "Libraries" field. If the ":" doesn't work, try using ";".

And now let's get down to serious business!

2. Available

8 PCB footprints and 6 associated symbols for getting started quickly:

- BOB-FT232R-MIN minimalist through-hole version with three pins: GND / TX / RX
- BOB-FT232R-TINY minimal version with I/O voltage available: GND / TX / RX / VCCIO
- BOB-FT232R-CABLE version equivalent to the FTDI USB-to-TTL cable, through-hole, six pins
- BOB-FT232R-WIDE complete reverse side of the bridge, through-hole, seven pins
- BOB-FT232R-EDGE connectors on both sides, through-hole and piggy-back mounting
- BOB-FT232R-FULL complete through-hole version (just in case)

3. Use

In EagleCAD select the required symbol, choose one of the associated PCB layouts, where applicable, click on *add* to add the symbol to your circuit diagram.

The layout corresponding to your choice will then be available in the board design window, all that remains for you to do is route it correctly.

Software peripherals

1. Microsoft Windows

To access the bridge via a COM port (in the good old-fashioned way), FTDI offers a COM port emulation driver (Virtual COM Port Driver) available from the address www.ftdichip.com/Drivers/VCP.htm.

You just have to install it for the bridge to be accessible via a COMx.

To emulate a terminal, we recommend TeraTerm or HTerm.

You can also install and use the D2XX drivers in order to directly access the FT232R core, but that's a whole other story...

2. Linux (core version 2.6.31 and above)

Linux cores in versions 2.6.31 or above incorporate the latest virtual port emulation drivers for the FT232R (ftdi_sio modules). There is nothing to install, and this probably holds good for older versions of the core too.

The bridge is accessible via the peripheral /dev/ttyUSBx.

To emulate a terminal, we recommend GTKTerm or HTerm.

3. MAC OSX

in order to access the bridge on your MAC, install the COM port emulation driver (Virtual COM Port Driver) offered by FTDI at the address: www.ftdichip.com/Drivers/VCP.htm.

The bridge will be accessible via the peripheral /dev/tty.usbserialx.

It is possible to emulate a terminal using the program *screen* supplied with MAC OSX.

The ferrite beads protect the circuit against electromagnetic interference from the USB cable.

Capacitors C3–C6 decouple the IC power supply.

The two LEDs light when data is being transferred over the serial link seen by the module (host). If the device connected to the bridge transmits a byte to it, the RX receive LED lights. Conversely, when a byte is sent from the computer via the bridge, it's the TX transmit LED that lights.

The circuit doesn't include any fuses, because I'm not sold on polyfuses (resettable chemical fuses) — the only plausible solution here. They reduce the voltage available, which, in my various experiments, dropped below 5 V. I prefer leaving all the voltage and power available to your circuit, which will come in very useful if, for example, you are thinking of including a 5 V/500 mA battery charger further down the line. So it's up to you to protect your circuit if necessary with fuses or power supply monitors downstream of the bridge.

The 5 V rail comes directly from the USB connector at the front of the board, and offers a maximum of 500 mA for a USB 2.0 port (150 mA for a USB 1.x). The 3.3 V rail is provided by the FT232R: take care not to draw more than 50 mA from it. Above this limit, the only signals you'll get will be smoke signals.

Did you notice JP1? It's the jumper printed on the board, and requires a solder bridge to configure the input/output signal volt-

ages: 3.3 V or 5 V. This needs to be set correctly **before** using the board: using a tiny blob of solder, connect the central contact to one of the two contacts on either side; the selected voltage is printed on the PCB: 5 V on the USB connector side, 3.3 V on the other. Above all, short only one contact at a time, as only one voltage is possible. Any other configuration would be fatal.

The simplest way of using the module consists in soldering a 3-way 2.54 mm (0.1") pitch pin header to the GND/RX/TX signals opposite the USB connector. In this way you will obtain a USB–UART bridge that is simple, effective, and can be used in almost all circuits.

Then, depending on your needs, you can use a larger pin header, to have access to more signals: the track layout at the rear of the board lets you obtain the equivalent of an FTDI cable [1][2][3]. Along the sides, you'll find the other signals and the power for the FT232R, accessible via copper contacts on both sides of the board and on the plated edge (!), in the 15.24 mm (0.6") wide DIP18 format. This way, you can solder two straight pinheaders there and use it within other circuits, on a test board, or plugged into a PCB-mounted socket.

You can also solder the bridge board-to-board, piggyback fashion, directly onto the circuit with which you wish to use it.

You'll be able to quickly and easily incorporate this module into your circuit thanks to an EagleCAD library available on the article web page [4]. You'll also find there a detailed but condensed data sheet — an essential companion when designing and debugging your application.

COMPONENT LIST

Resistors
R1,R2 = 270Ω (0603)

Capacitors
C1 = 10nF (0603)
C2, C3 = 47pF (0603)
C4 = 4.7µF 6.3V (0603)
C5, C6 = 100nF (0603)

Inductors
L1, L2 = ferrite bead >30Ω @ 100 MHz / 1.5 A (0603)

Semiconductors
IC1 = FT232RQ
D1 (TX) = LED, green (0603)
D2 (RX) = LED, red (0603)

Miscellaneous
K1 = USB mini-B connector
18-pin (2x9) pinheader, right angled
Pinheader

200 % of real size

Even an experienced electronics technician with good eyesight and equipped with suitable tools (in particular, a hot-air soldering iron) will also need to have confirmed masochistic tendencies to set about— and above all pull off — the construction of their own prototype. And those with trembling hands had better steer well clear!

The task is so tricky, in fact, that we're offering (and recommending) the circuit pre-assembled, ready to use, with the various extension connectors as a bonus. See the product page for further details [4].

(110553)

Internet links

[1] *USB-to-TTL Serial Cable*, June 2008, and associated cables in 5.0 V and 3.3 V versions (Elektor ref. 080213-71 & 080213-72): www.elektor.com/080213.

[2] *USB <-> RS-232 cable*, July/August 2008 (Elektor ref. 080470): www.elektor.com/080470

[3] *USB/TTL serial cable, extension & supplement*, July/August 2010 (Elektor ref. 100007): www.elektor.com/100007

[4] BOB-FT232R: www.elektor.com/110553

[5] Web page for FT232R IC: www.ftdichip.com/Products/ICs/FT232R.htm
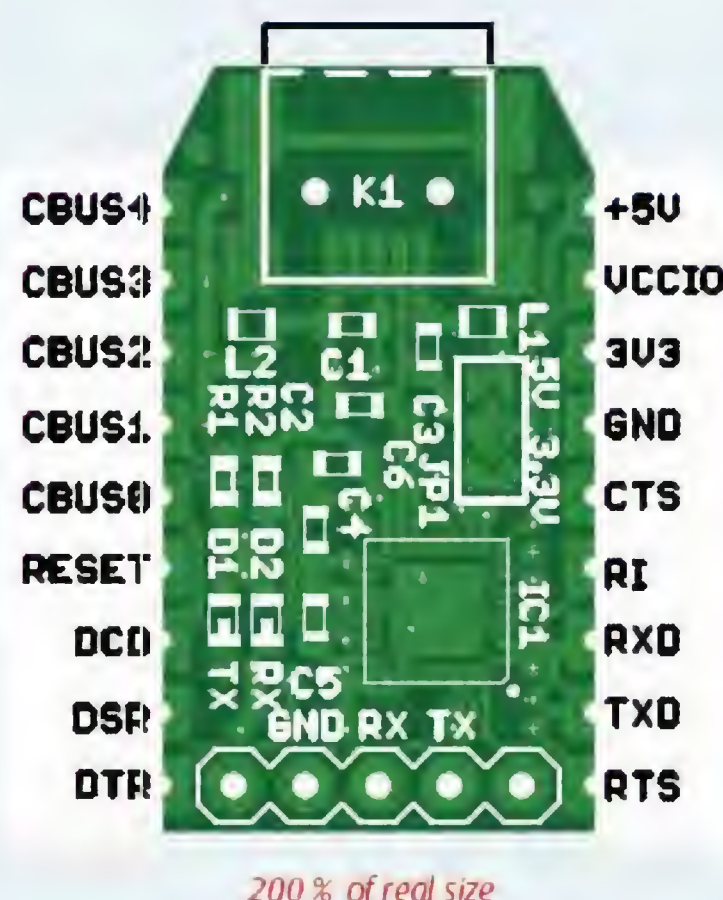
Here Comes the Bus! (7)
A simple application protocol
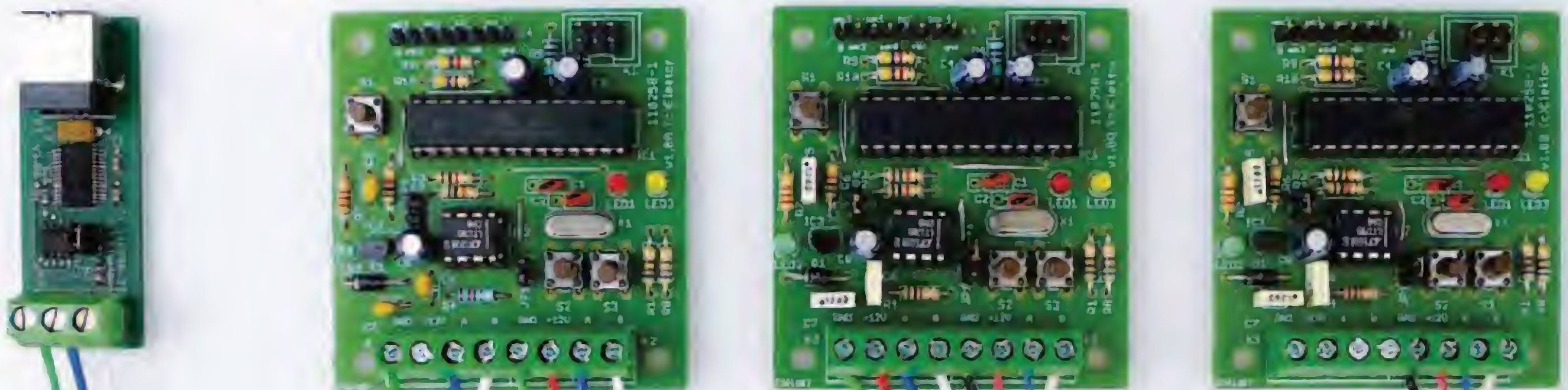
After a brief pause for the summer holidays our bus resumes its normal timetable. In this article we describe a simple protocol that allows up to four set-points and corresponding instantaneous values to be transmitted simultaneously. The result is ideal not just for home automation applications, but more generally for measurement and control. Also, for the first time, we look at programming in C with AVR Studio.

By Jens Nickel (Elektor Germany Editorial)

In previous instalments in this series we presented a simple frame protocol which allowed a payload to be sent from a transmitter to an addressable receiver device. Take a look at **Figure 1** to remind yourself of the details. A message in the Elektor Message Protocol essentially consists of sixteen bytes, where byte 0 always has the value AA_{hex} (170 decimal) for synchronisation purposes. If both bits 7 and 6 of the following mode byte are zero, bytes 2 to 5 are used for addressing. Since bytes E and F are reserved for the optional checksum, it is possible to use up to eight bytes for the payload.

Previously we also looked at a simple way to regulate the traffic on the bus ('hybrid mode'). Nodes that have a message to send on a regular basis (such as sensors)

are interrogated in turn by a scheduler. Between these times are the so-called 'free bus phases' during which nodes are allowed to speak without specifically being asked. Collisions can occur during this period and so 'non-scheduled messages' (in other words, messages transmitted during the free bus phase) must be acknowledged by their recipient. This is done by sending an 'acknowledge message' back to the sender.

Non-scheduled messages are particularly required when a node needs to communicate something as a result of an external event but could otherwise perfectly well remain silent. In the interests of efficiency it is better not to poll such a node regularly. An example of such a node in the area of home automation might be a light switch. Equally, a sensor that only needs to report when a value has gone outside pre-

set thresholds would fall into this category, an example of such a sensor being a water level detector.

Sub-nodes

Hybrid mode is particularly useful when a node both has to be interrogated regularly **and** needs to send event-triggered messages during the free bus phase. Think for example of a temperature sensor that regularly reports the current temperature reading but which also monitors these readings against a threshold. This possibility was not explicitly covered in the previous instalment: in the demonstration software presented there I drew a strict distinction between 'polled nodes' (perhaps better described as 'scheduled nodes') and 'free bus nodes' [1]. An on-the-ball reader immediately suggested to me that it was possible for a node to have both behaviours simulta-

neously. Francis Stevenson also put forward the suggestion that such a sensor node should always first report the fact that a threshold value has been crossed, whether during a regular interrogation or during the free bus phase. This is a good idea, especially for particularly urgent messages. Francis and I also discussed the possibility of being able to have more than one 'device' on the same physical node. A node board would then respond to more than one address, and hardware costs could be reduced. The firmware running in the microcontroller must make sure that messages are correctly routed to the sub-units within the node. The same basic principle was used in the demonstration software in the last instalment [1], where the PC simultaneously took on the roles of scheduler (address 0) and master (address 10).

More channels

However, if we only have a couple of simple sensors and/or actuators on a single node, the splitting into devices each with their own address is not necessary. Indeed, it would be inefficient if each sensor had to send a separate message from its own transmitter address to the master to communicate just one value. A better approach in such cases is to use 'channels' (hello DMX!). Since we have eight payload bytes available in a message, we can easily send four temperature values (each consisting of two bytes) at the same time without extra overhead. That fits very neatly with our experimental node hardware, which has four ADC inputs available on header K4.

Which bytes in the payload correspond to which channel (and hence to which sensor) is then simply a function of their position: the value for channel 0 is sent first, followed by that for channel 1, and so on (see **Figure 2**). Using the same idea we can also control four actuators using a single message, always assuming, of course, that each control value can be expressed in two bytes. In the demonstration software in the previous instalment we used two bytes to communicate one of the ten-bit values read from the microcontroller's ADC. We packed the lower seven bits of the result into one payload byte and the upper three bits into
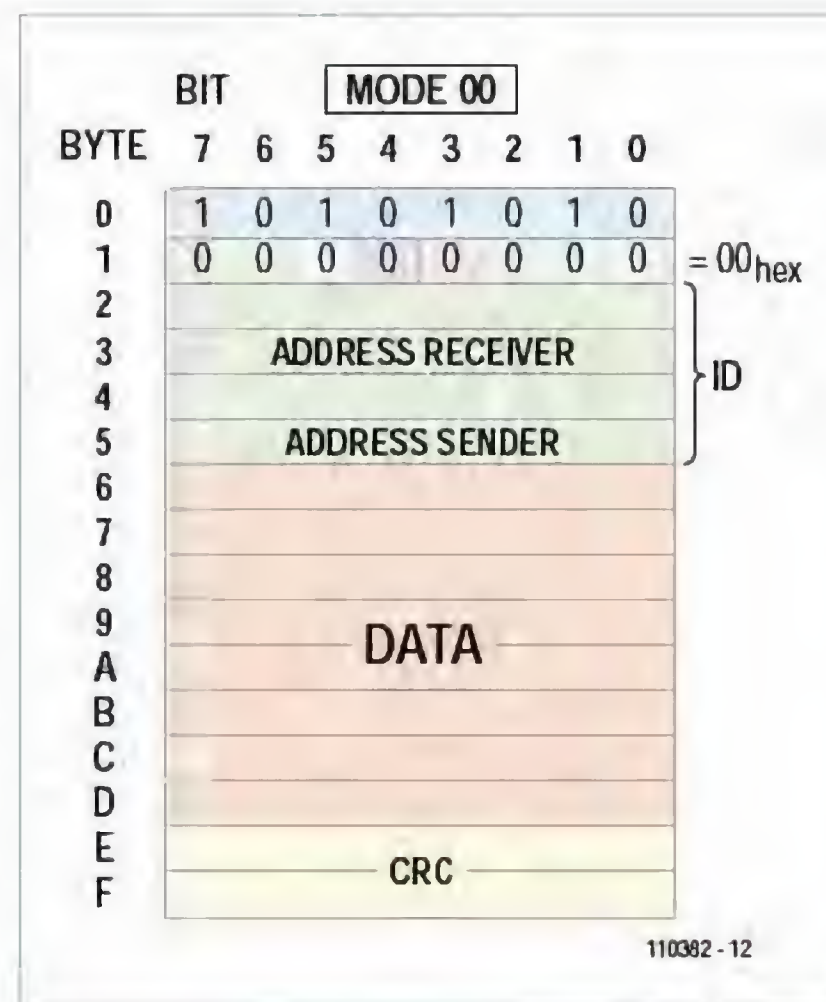

Figure 1. The Elektor Message Protocol defines only where the payload is located within the frame [6].
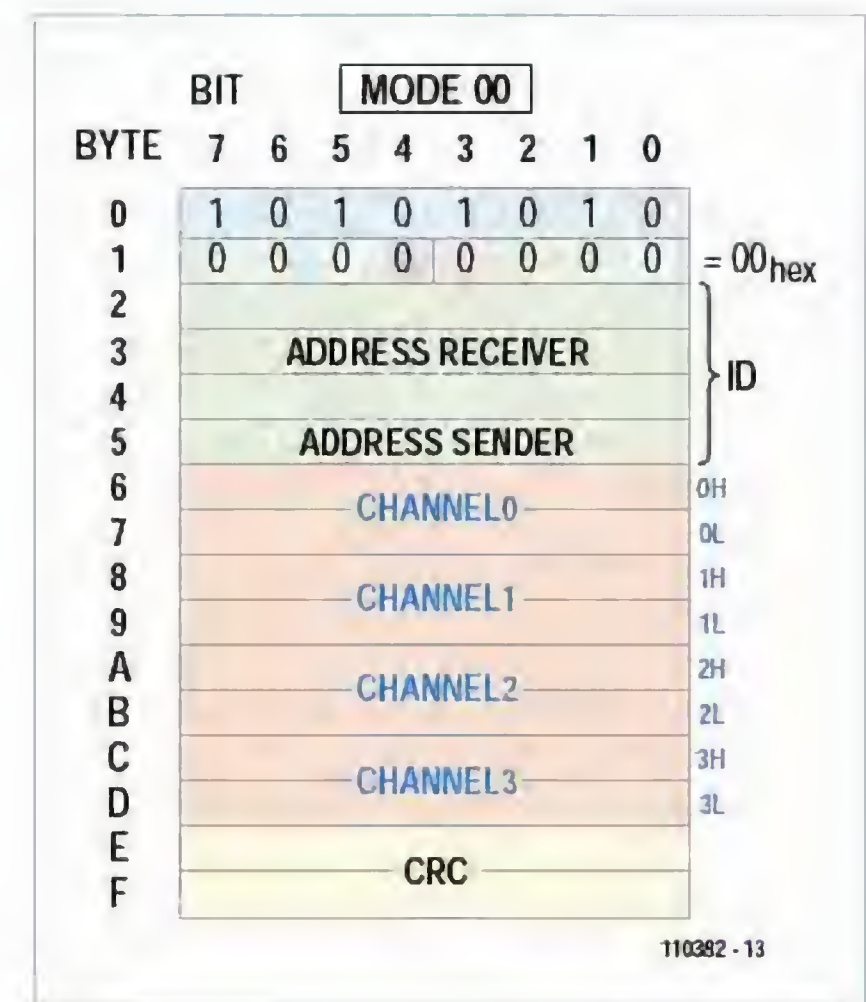

Figure 2. Our application protocol divides the eight payload bytes into four consecutive byte pairs. Values are transmitted in one of four channels (for sensors and actuators attached to a node).

the next payload byte. This has the advantage that the value AA_{hex} can be prevented from ever occurring as a payload byte, which would otherwise confuse our simple synchronisation system. We can use the same trick for each of our channel values: and already we are half-way into defining our application protocol!

The Elektor Application Protocol

So we need an application protocol mutually understood by the nodes on the bus (both sensors and actuators) and which will allow easy expansion to accommodate new hardware. So that we do not have to reinvent the protocol every few months, we have kept the Elektor Application Protocol relatively simple and yet also flexible, fulfilling the following requirements as a minimum.

- Transmission of ten-bit values plus sign, either a reading from a sensor or, in the other direction, a control value to an actuator.
- The option to use twenty-bit values plus sign, for which we need a four-byte-per-channel mode.

- Setting of units and scaling factors for smart sensor nodes.
- Setting of measurement interval for sensor nodes.
- Setting of multiple thresholds.
- Notification of above- or below-threshold alarms.
- Configuration and calling-up of default presets (for actuators).
- Distinguishing between an acknowledge message, which contains the received values sent back to the transmitter for checking, and the original message. (We have already implemented this feature in the software presented in the previous instalment.)

The protocol should also not be limited either to use with hybrid mode or to home automation applications. It should, for example, be perfectly suitable for remote interrogation of a meter or other point-to-point applications (in which collisions can be completely avoided).

Control with ten bits

To make this article more than just a bald datasheet we will look at the functions

Using AVR Studio and BASCOM in parallel

With the double summer edition of Elektor put to bed I set to work on fulfilling my promise that we would be presenting some C code for the system. The newest version (5.0) of the AVR Studio development environment includes an integrated C compiler (AVRGCC) behind its powerful and user-friendly interface; it is free to download (after registration) from the Atmel website [3].

The first problem was to get the AVR Studio environment to talk to the AVRISP mkII programmer. Although I had already installed the necessary driver when installing the development environment itself, at first things did not work properly. The problem was that the programmer was bound to the libusb driver which I had installed for use with BASCOM. Uninstalling the libusb driver solved the problem, and I was able to program devices from AVR Studio without further difficulty, simply by plugging the programmer into a USB port. The screenshot shows how things appear in the Windows 7 Device Manager when correctly set up. Of course, I wanted to use BASCOM at the same time, and this can also be made to work: the libusb driver has to be installed as a so-called 'filter driver'. There is a discussion of how to do this at [4].

When setting up a new project in AVR Studio it is necessary to specify the target processor. Fortunately there is not a lot else to configure. A tap of F7 (or 'Build Solution' in the menus) creates a hex file from the source code and any referenced libraries. The programmer is thus insulated from the difficulty of writing a makefile to control the build process.
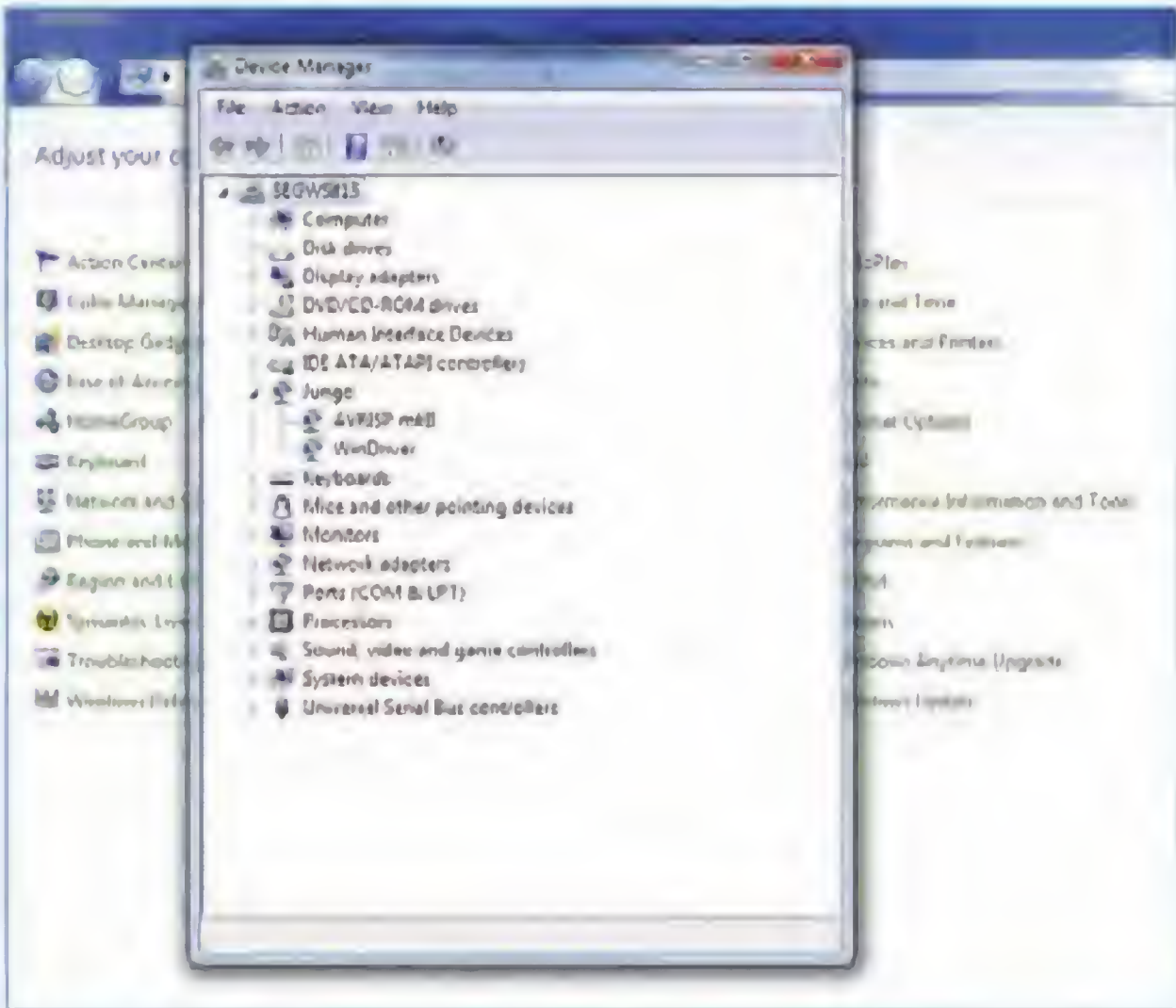
To flash the target device click on the button with the lightning icon. In the programmer window that opens first click on the 'Apply' button towards the top (with the settings Tool=AVRISP mkII, Device=ATmega88, Interface=ISP). The 'Program' button is located under 'Memories'. The path to the hex file should first be entered into the combo box above.

listed above step by step.

We start with the communication of ten-bit values which, as we mentioned above, are divided between the two bytes forming channel 0. We call the two parts 0H (for 'high') and 0L (for 'low'). As **Figure 3** shows, bits 0H.7 and 0L.7 (the most significant bits of the two payload bytes) are always zero, so that the byte value AA_{hex} cannot occur. This uses up 50 % of the possible values for each byte, but we can always use the remaining possible values for special functions later if we wish.

We reserve bit 0H.3 for the sign of the value: 1 representing negative, 0 positive. The data bits D9 down to D0 are then packed as described above. Three bits are left over, which we use as follows.

Bit 0H.6 determines whether two bytes or four bytes are being used for the channel. The four-byte mode will be used later to send more precise data values and for certain special functions.

Bit 0H.5 says whether the data value is a set-point or an instantaneous reading (1 indicating set-point, 0 an instantaneous reading).

Bit 0H.4, when set, indicates that this is an acknowledge message.

Figure 3. The two bytes forming one channel in more detail. We can transmit values with up to ten bits of precision, plus a sign bit.

As an example, consider a Venetian blind which a home automation master controller wants to set to a 30 % closed position. (For this example ten bits of precision are more than enough!)

The master and the blind controller must have agreed beforehand on how the 30 % closed position is represented numerically: we will look at scaling factors in a later instalment. Let us suppose that we encode the percentage directly as an integer, and that we can communicate with the blind controller on channel 0. In bytes 6 and 7 of the Elektor Message Protocol packet the master will then send the following two bytes:

0-1-1-0-0-0-0-0   0-0-0-1-1-1-1-0
($00011110_{bin}=30_{dec}$)

The blind controller replies with an acknowledge message, having the acknowledge bit set:

0-1-1-1-0-0-0-0   0-0-0-1-1-1-1-0

An intelligent blind controller could of course determine the instantaneous position of the blind and report this value. It would in any case be wise to have it report an instantaneous value of 30 % when the process of moving the blind has completed:

0-1-0-0-0-0-0-0   0-0-0-1-1-1-1-0
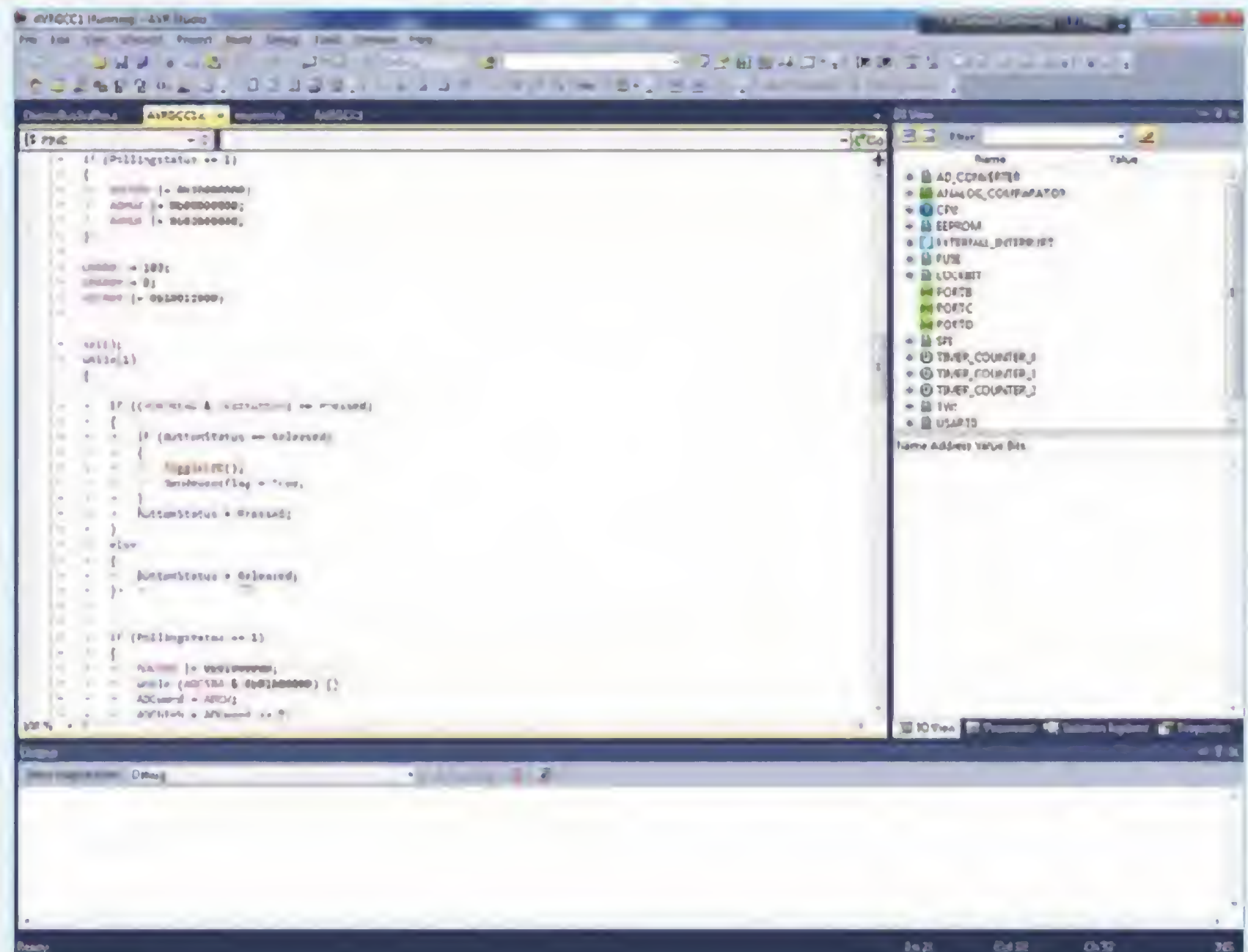
First experiments in C

I found the Internet a great ally in my first experiments in embedded C programming. The pons asinorum was to get some LEDs flashing; then I moved to to reading values from the ADC and outputting a few bytes onto the bus using the microcontroller's UART, which I read back into the PC and displayed using the Terminal.exe terminal program. It is important to be wary when copy-and-pasting programs from the Internet as you can find code for a range of different AVR microcontrollers which cannot always be used directly on the ATmega88 without checking first against its datasheet [5]. For example, the naming of registers can vary between different microcontroller types: UDR (the register that accepts bytes to be transmitted and holds received bytes) is called UDR0 on the ATmega88.

A particularly nasty trap is the naming of the interrupt vector which is used to specify the routine to be called when a character is received by the UART. Many Internet code examples give the incantation 'ISR(USART_RXC_vect) { ... }', the use of which has the unfortunate effect of hanging the microcontroller. After a good hour of head-scratching I discovered that the correct form for the ATmega88 is 'ISR(USART_RX_vect) { ... }'.

Anyone coming from the world of BASCOM or BASIC more generally should be particularly aware of the following types of error which the compiler will not always complain about. One schoolboy error is to confuse a doubled equals sign (used to indicate an equality comparison) with a single equals sign (used to indicate an assignment). The C language is also absolutely strict when it comes to case sensitivity, both in variable names and in keywords such as 'if'. A pair of brackets is essential after a function name to indicate a call to that function (e.g., 'ToggleLED()'), and a misplaced semi-colon can lead to all sorts of surprising error messages. It is a good idea to check over the program syntax carefully before setting the compiler loose on your code.

The next exercise was to translate the BASCOM demonstration software into C. The result can be found on the project web pages [2]. To make comparison as easy as possible, I have tried to adhere to the structure of the original code as far as possible. There are of course many opportunities for optimisation, and old hands at C are encouraged to send in improved versions!

It is apparent from a side-by-side comparison that programming in C requires getting closer to the hardware. BASCOM hides a lot of the nitty-gritty behind commands like 'Start ADC', 'Enable Urxc' and 'Printbin'. However, having to learn what the microcontroller's registers do is not necessarily a disadvantage, and the extra programming effort can be more than outweighed by the advantage of having a set of reusable, made-to-measure routines. As we de-

velop the firmware further these subroutines will be packaged into a small library.

A further important point for beginners in C is that ports and other registers (for example for the ADC and UART) must always be addressed as complete bytes. If only one bit is to be set, care must be taken to preserve the others. This can be done using a logical OR operation:

`PORT = PORT | Bitmask;`

or more concisely

`PORT |= Bitmask;`

To clear a bit, use a logical AND operation with an inverted bit mask:

`PORT &= ~Bitmask;`

I have used directives such as

`#define TestLED 0b00010000`

to create bit masks and port names corresponding to the LEDs, buttons and port pins. This allows a statement of the form

`PORT |= TestLED;`

to be used to light the test LED. Much code found on the Internet uses a slightly different approach, writing

`#define TestLEDbitposition 4`

`...`

`PORT |= (1 << TestLEDbitposition);`

where the expression `1 << TestLEDbitposition` generates the required bit mask by shifting the value 1 left by the appropriate number of places.

Finally the master confirms that this value has been received:

0-1-**0-1**-0-0-0-0   0-0-0-1-1-1-1-0

Perhaps this communication scheme might only be implemented in a trimmed-down form, but the example nevertheless gives a good demonstration of the use of the set-point and acknowledge bits.

Demonstration software

Again, along with this article we bring you example software for the PC and for the ATmega88 microcontroller used in the experimental node. It is a modified version of the demonstration software from the previous article, allowing all three nodes to report the status of their test LED to the PC software, and one of the nodes to report the voltage read on its ADC0 input. The software, as usual, is available for download from the *Elektor* website as a zip file [2]; for comparison, the previous version is available at [1].

Channel 0 is used to transmit the ADC reading (only from node 2); the LED status information is transmitted on channel 1. One byte in the EEPROM (at address 006) determines whether the node sends ADC values or not; the corresponding variable in the

Figure 4. Screenshot of the PC-based software showing the status of the test LEDs on three nodes and the ADC0 value from node 2.

code is called 'Devicemode'. This byte must be set to the value 01 in node 2: this can be done manually using BASCOM. The variable 'Pollingstatus' has been renamed as 'Scheduled' in the code, which better conveys the

idea of whether the node is a 'scheduled node' or a 'free bus node': these are mutually exclusive in this version of the software. For the first time we also include firmware written in C using the new AVR Studio 5.0: the text boxes give the low-down. BASCOM users will also find it worthwhile to see how things are done in C code.
More next month!

(110382)

**What do you think? Feel free to write to us with your opinions and ideas.**

Internet Links

[1]   www.elektor.com/110258

[2]   www.elektor.com/110382

[3]   www.atmel.com/dyn/products/tools_card.asp?tool_id=17212

[4]   http://avrhelp.mcselec.com/index.html

[5]   www.atmel.com/dyn/resources/prod_documents/doc2545.pdf

[6]   www.elektor.com/110012

USB Audio Adapter
an add-on USB input
for external D/A converters

Karl Köckeis

You need a good quality external D/A converter and amplifier if you want to play audio files on your computer in high fidelity. The missing link in the chain is often a digital S/PDIF output from your computer. The low cost solution suggested here modifies an off-the-shelf analogue USB audio card and provides the S/PDIF connection almost for free.
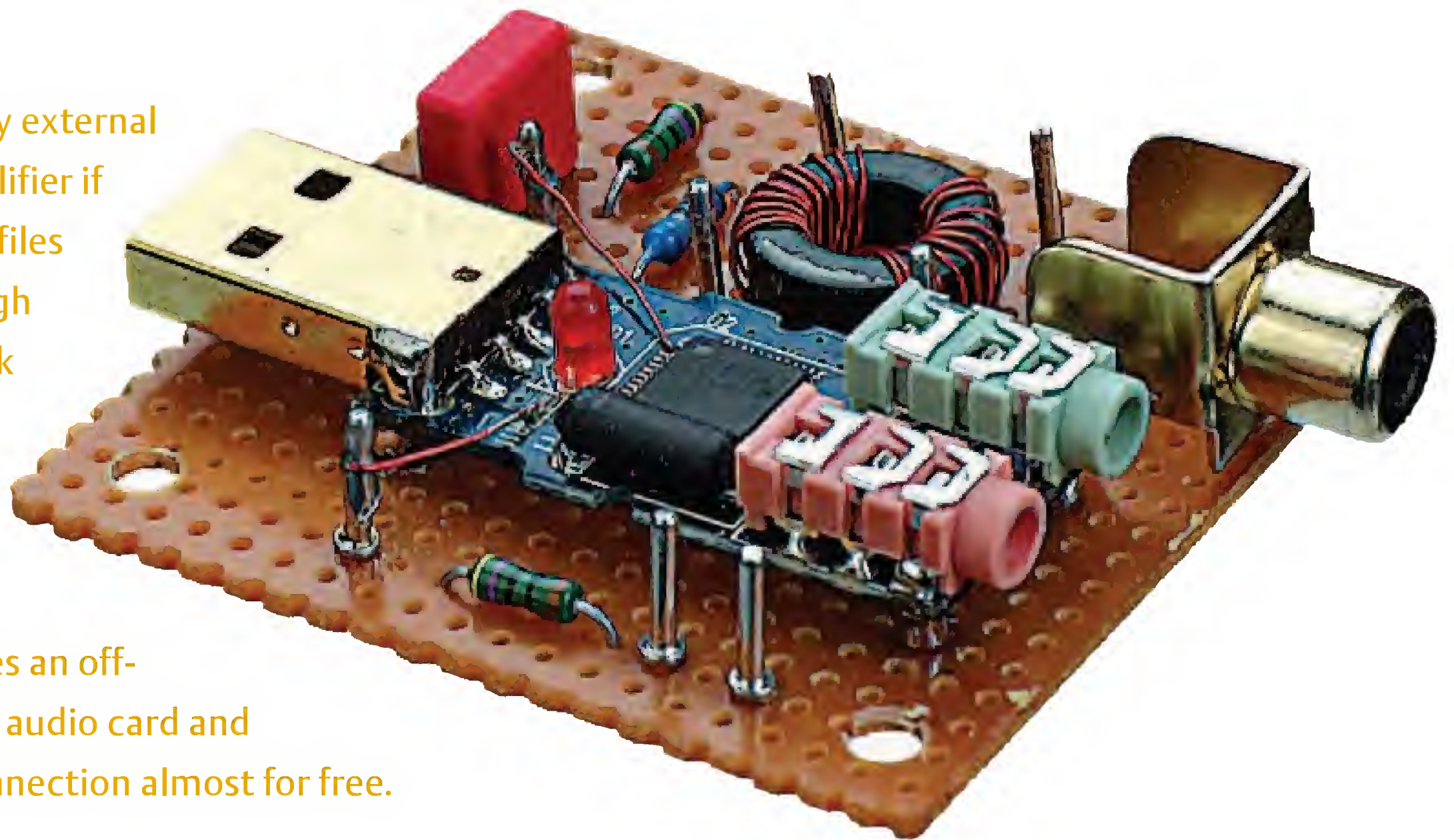
Music collections stored on a PC are easy to organise and catalogue and when it comes to playback an external audio D/A converter offers best quality reproduction. It is of course necessary to link the two pieces of equipment together. The most common interface for digitised audio signals is the S/PDIF standard (Sony/Philips Digital Interface) but usually only higher spec PCs with an additional sound card provide such a connector and even then it is normally hidden away at the back of the machine.

A more convenient way of outputting digital data is via a spare USB port; these have been fitted as standard on PCs for more than a decade now so even quite old machines and laptops should be well equipped.

An external USB audio adapter with S/PDIF output can now be used between the PC's USB port and the digital input of the D/A converter but it would be more convenient if the D/A converter had a built-in USB input. The converter described here provides a simple, low cost USB input port which can be fitted into an existing external D/A converter. An example is shown on the author's
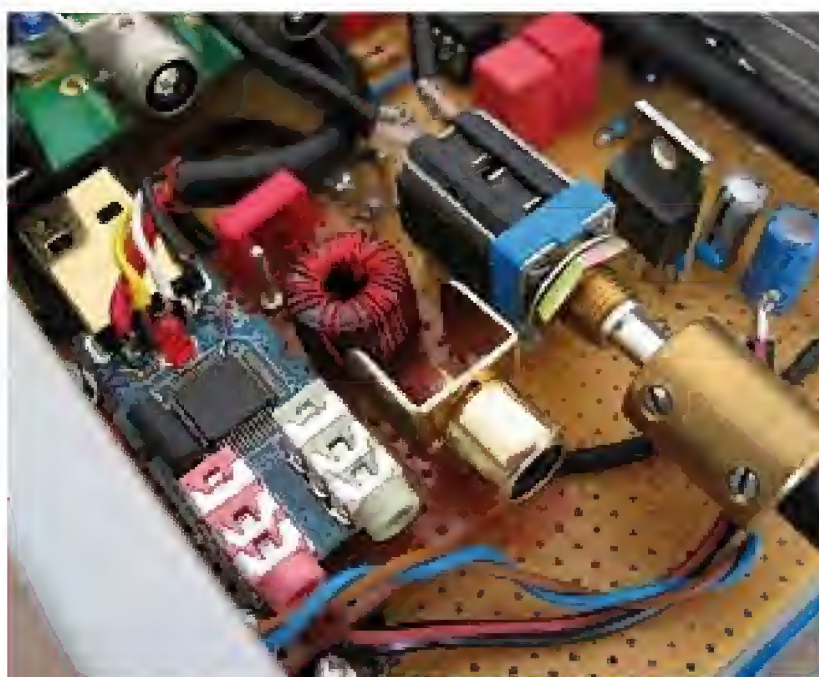
Figure 1. The converter offers a very simple method of fitting a USB input to an external D/A converter. (The author's audio equipment is shown here, see [1])

home page [1] (see **Figure 1**). Those of you considering building a high-quality D/A converter or who already have one available may therefore consider incorporating this circuit into the equipment. Two examples of audio D/A designs can be found in Elektor November 1999 and July/August 2002 [2].

Function

Starting form scratch, if you were to design a circuit to produce a digital S/PDIF signal from the PC's USB port you would need hardware to provide an interface to the the outside world. Making the necessary modifications to this adapter simplifies the task considerably; the adapter already comes in a neat plastic enclosure, including the USB plug and is recognised by Windows Vista and XP as an audio device without the need to write or run any additional software (A driver for Windows 98 is included).

Once plugged in, most machines will automatically route audio through this adapter but it can be switched back through the internal sound card by changing the audio device options (**Figure 4**). For our application here it is only necessary to make connections to the correct pins on the chip and route the signal through a suitable digital

Figure 4. The adapter is recognised as a soundcard chip once it is plugged into a USB port on the PC. Audio can be routed through the internal sound card, through the adapter or through any additional audio equipment.

Figure 2. The converter is formed from a modified and recycled 'Ultra Portable Audio Card' from Speed-Link.

Figure 3. In addition to the soundcard chip the adapter contains other useful components like the USB connector.

USB port, some logic and S/PDIF interface hardware. A considerable amount of software would also be necessary to communicate with the PC operating system and handle the data flow. The majority of these features are already integrated into LSI sound chips that are fitted inside external USB audio cards.

These cards are designed to provide additional analogue audio input and output via a USB port. In particular the 'Ultra portable sound card' from Speed-link (**Figures 2 and 3**) is of interest here, it contains the C-Media [3] CM108 audio processor chip which along with the analogue input and output signals contains a built-in S/PDIF output. In this particular product the S/PDIF output signal has no connection to

output circuit, the analogue input and output signals are still available for use but are not required here.

Coax or optical?

The digital signals can be conveyed optically using a fibre optic cable or electrically using a coax cable. The majority of professional equipment manufacturers show a preference for coax connection, the cabling is cheaper and the cable can be run round tight bends without attenuating the signal. It is unlikely that any detectable difference in the reproduced audio can be put down to the type of medium used. The modification suggested here provides a coax connector, an optical connection could also be provided if you prefer and a circuit diagram showing the hardware is given on the CM108 data sheet.

Transformer coupling

Wideband transformer TR1 couples digital signals to the external D/A converter while suppressing any unwanted noise and providing galvanic separation. In order to convey the digital signal without distortion the transformer needs to have wideband characteristics. The choice for the constructor is whether to use an off-the-shelf transformer (often difficult to find a reliable source) or go the wind-it-yourself route (requires a little time and effort to make). A suitable off-the-shelf transformer is the PE-65612 from Pulse which offers a transmission rate of 1 to 7 Mbps with a pulse rise time of 25 ns and an isolation voltage of 2 kV.

An alternative homebrew wideband transformer can be made using an Amidon ferrite core. The toroidal core type FT 50A-77 has an A_L value of 1100 nH/n² and is suitable for use in the frequency range of 0.5 to 50 MHz. Two windings, each of 10 turns are made on the toroidal core using 0.5 mm enamelled copper wire (**Figure 5**).

Lab testing

The prototype was put through its paces in the Elektor lab by our audio specialist Ton Giesberts. The converter generally performed well with many different audio D/A converters. Unfortunately using the audio DAC featured in a 1992 edition of Elektor, Ton encountered a synchronisation problem and was unable to identify the source of the trouble. Any reader out there who encounters similar problems with this cir-
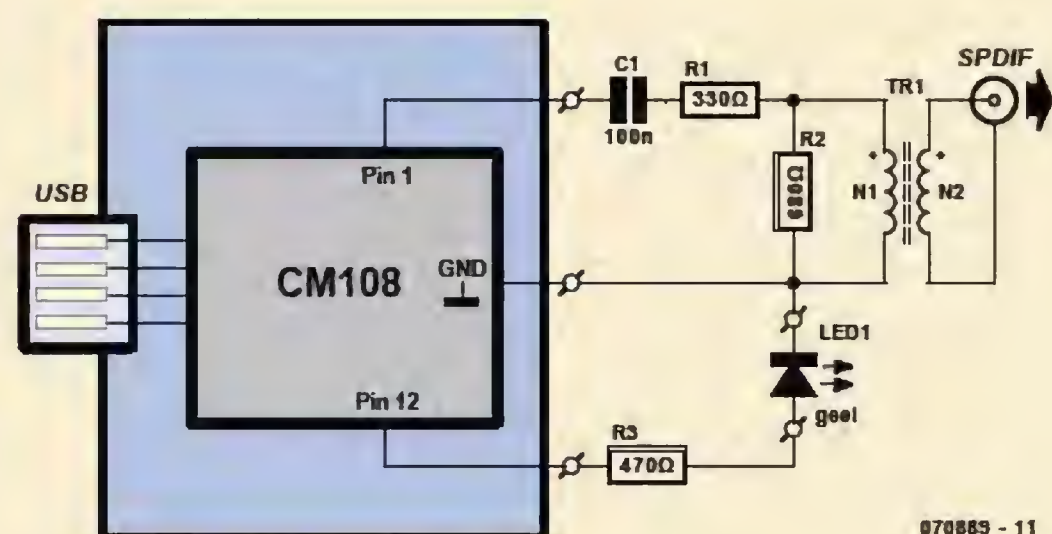
cuit is invited to post comments on the Elektor forum ('Audio' topic); reports of success are also most welcome there!

Even though the data sheet for the CM108 mentions just two sampling frequencies (44.1 and 48 kHz) 32 kHz and 96 kHz are also no problem for the homebrew converter (tested with Windows).

COMPONENT LIST

Resistors
R1 = 330Ω
R2 = 680Ω
R3 = 470Ω

Capacitor
C1 = 100nF

Inductors
TR1 = PE-65612 (Pulse) or
    home made (2 x 10 turns on Amidon FT 50A-77)

Semiconductors
LED1 = standard LED, yellow

Miscellaneous
Cinch socket
Solder pins
USB audio adapter type UltraPortable Audio Card from Speed-Link

Figure 6. Most of the features are implemented in the adapter already so the circuit diagram of the modification is very simple.

Circuit

The majority of the circuitry for this design is already contained in the USB sound adapter so the add-on S/PDIF port shown in **Figure 6** is quite simple. The CM108 sound chip outputs the digital signal on pin 1, this is ac coupled to resistor R1 and then fed to the output transformer TR1. R1 and R2 define the transformer output impedance and signal level (0.5 V and 75 Ω). The coupling transformer has a turns ratio of 1:1 and provides galvanic separation between the PC and external D/A converter. The upper frequency response of the transformer is limited by the coupling factor between the two windings, the toroidal core gives the best possible coupling and the widest bandwidth. The ferrite core material is suitable for use with signals up to around 50 MHz. R3 is used to limit current to the optional status LED.

Figure 5. The homebrew transformer consists of two windings, each of ten turns of enamelled copper wire.

Construction

First it is necessary to prise the two halves of the Speed-link USB audio card apart using a thin screwdriver.

The adapter PCB can then be soldered onto the corner of the Speed link PCB

using four shortened solder pins. The components can now be soldered to the PCB and three wires used to make connections to the Speed link PCB. One wire from pin 1 of the 48-pin IC connects to C1; another links earth between the two PCBs and the third connects Pin 12 with R3 for the status LED.

The earth connection can be made alternatively using one of the solder pins already soldered to the Speed link PCB. Be careful not to make any solder bridges between pins of the IC, they are very closely spaced. The complete audio adapter can be mounted inside the enclosure of a D/A converter so the status LED can be fitted to the front panel and connected via short lengths of wire back to the PCB.

(070889-I)

Internet Links

[1]   Author's homepage: www.htfi.de

[2]   D/A converters featured in Elektor: www.elektor.com/
      magazines/2000/january/audio-dac-2000-(3).56175.lynkx
      www.elektor.com/magazines/2002/july/
      mini-audio-dac.55832.lynkx

[3]   Datasheet for the C-Media CM108:
      www.cmedia.com.tw/?q=en/USB/CM108

[4]   Homepage of Speed-Link: www.speed-link.com

About the author

After studying telecommunications at the technical college in Regensburg Germany Karl has been working in the telecommunications industry, amongst other activities he has been working in the field of software development. Since his school days he has been interested in audio engineering and admits that he has often been inspired by projects and articles that have appeared in Elektor. More recently he has found more time to spend on his favourite pastime: the development, construction and modification of audio equipment.

Compact Warning Flasher
Better safe than sorry

By Peter Lehmann (USA)

This little project should give peace of mind to bicycle riders pedaling in the dark without street lighting. At the same time it provides early warning to faster, motorised road users to pay attention and pass with care.

The project is primarily intended as a red warning flasher to be mounted on the rear of a bicycle, although it is sure to have many other uses. All parts used should be easy to find and the circuit is highly open to your experiments in terms of LED flash rate, LED types used, and so on.

How it works
In the circuit diagram shown in **Figure 1**, four identical ICs are seen: the "flying capacitor"-type DC voltage converter ICL7660 from Maxim. The ICL7660 (or its 'Maxim'-ised equivalent the MAX1044) can be configured to invert, double, divide or multiply a positive input voltage — see its datasheet at [1].

In the circuit discussed here, converter IC1 is conventionally configured as a negative voltage generator. The next one, IC2, functions as an SPDT (single-pole double throw)

switch providing or removing supply voltage alternately to converters IC3 and IC4 at about 0.5-second intervals. When converter IC3 is connected to power, flying capacitor C3 is alternately charged and then discharged through series connected LEDs D1 and D3 at about 1/20th of a second intervals. Converter IC4 is connected to power as converter IC3 is disconnected, and then operates in the identical active fashion as IC3 and associated components, i.e. D2 and D4 flashing with electrolytic capacitor C4 acting as the reservoir device in the charge pump.

Two series connected AAA alkaline batteries power the circuit with a nominal voltage of 3 V. When the battery voltage has diminished to less than 2.4 V, the intensity of the LEDs decreases markedly. Fortunately, with respect to a battery voltage of 3.0 V (or fully charged), the average current drain by the

circuit equals 19 mA. So the rate of discharge equals about $C/60$ and the battery voltage drops to less than 2.4 V only well towards the end of the discharging cycle. Ah, yes, C is the nominal capacity of the battery expressed in mA/hour.

Circuit board, box and bicycle
A simple circuit board was designed by the author to accommodate the parts. The component layout and associated copper track are shown in **Figure 2**, as well as provided as files you can download from the project web page [2]. All parts are through hole and mounting and soldering should be a walk in the park provided you work carefully. Due attention should be given to the polarization of the electrolytic capacitors in the circuit.

The proposed enclosure for the project is the Velleman G203 sealed polycarbonate

---

Note. Readers' Projects are reproduced based on information supplied by the author(s) only.
The use of Elektor style schematics and other illustrations in this article does not imply the project having passed Elektor Labs for replication to verify claimed operation.
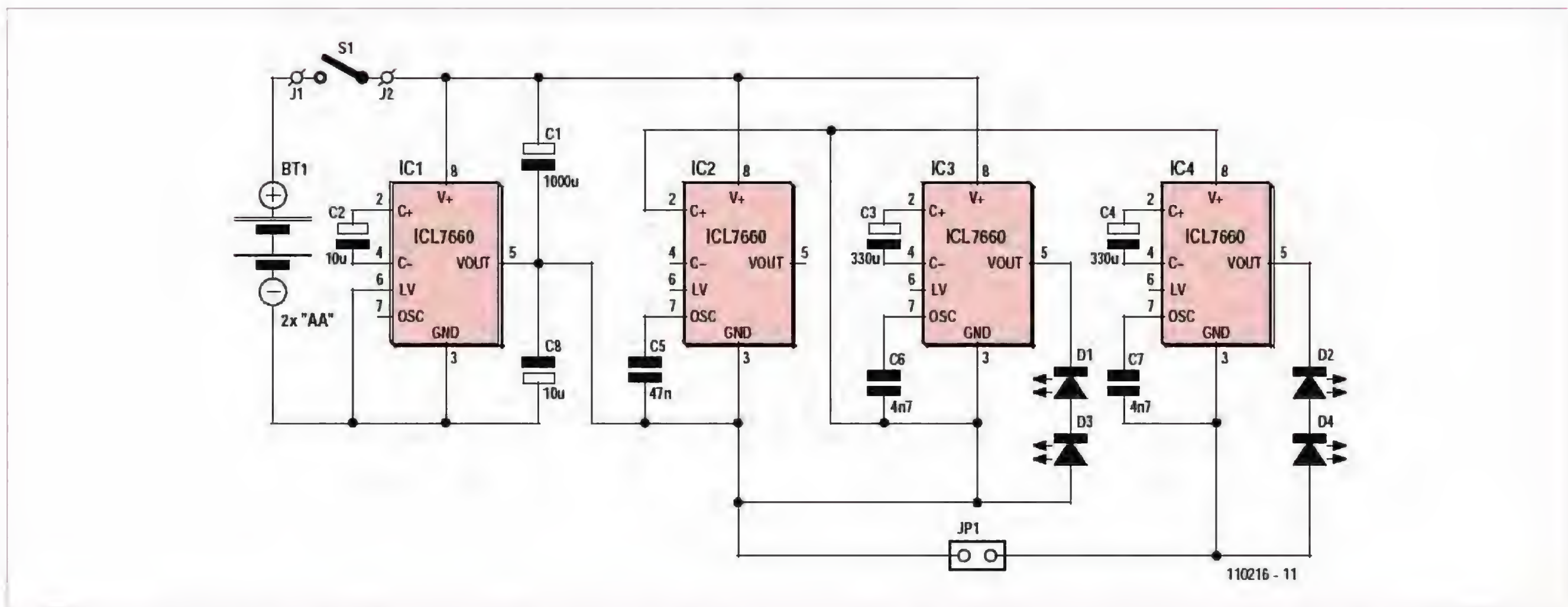
Figure 1. The schematic diagram of the flasher is heavily dominated by four ICL7660 20 mA charge pump ICs.

box with a clear cover. The box measures 115 × 65 × 40 mm (4.5 × 2.5 × 1.6 inch). The component layout includes rectangles at the four corners of the PCB indicating areas to be removed allowing it to fit in the G203 box. The four dots on the PCB indicate the position of mounting holes.

Red LEDs D1-D4 are 10 mm size for two reasons. Firstly, the 10 mm size has greater visibility than 5 mm, 5 mm being almost universally the size of the LEDs of commercially available warning flashers for bikes. Secondly, 5 mm LEDs would look lost in the Velleman G203 box.

A waterproof pushbutton switch was fitted on the sidewall of the bottom half of the box next to IC2 and capacitor C1. The photograph shows one possible method of attaching the flasher to a bicycle. This involves securing a galvanized steel corner brace to the mounting flange of a single bolt seat-post through a hole at one end of the brace. The other leg of the brace is attached to the G203 box by means of a bolt through one of the two pre-existing mounting holes of the G203 box.

(110216)

Caution. This circuit may not be road legal depending on traffic laws and safety regulations in your state or country.
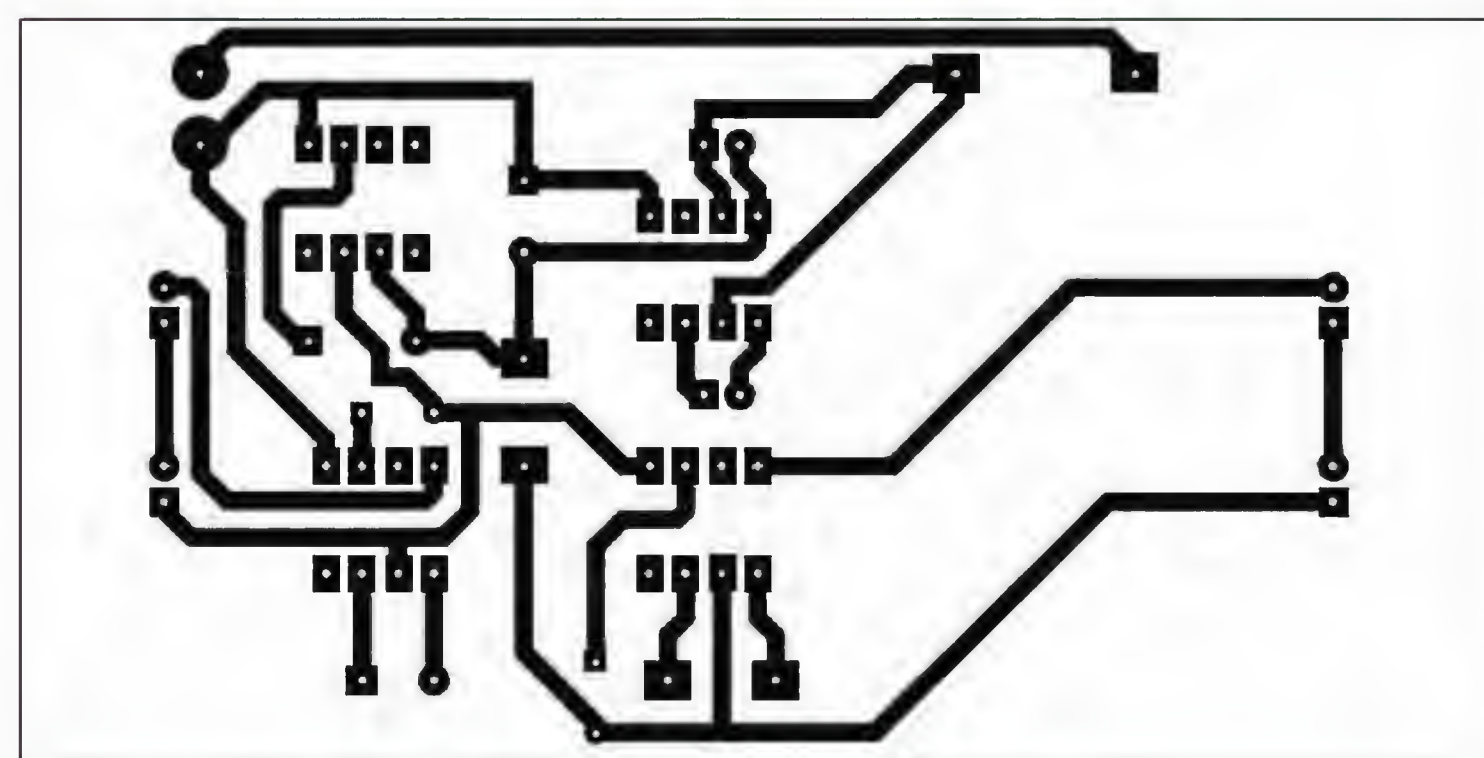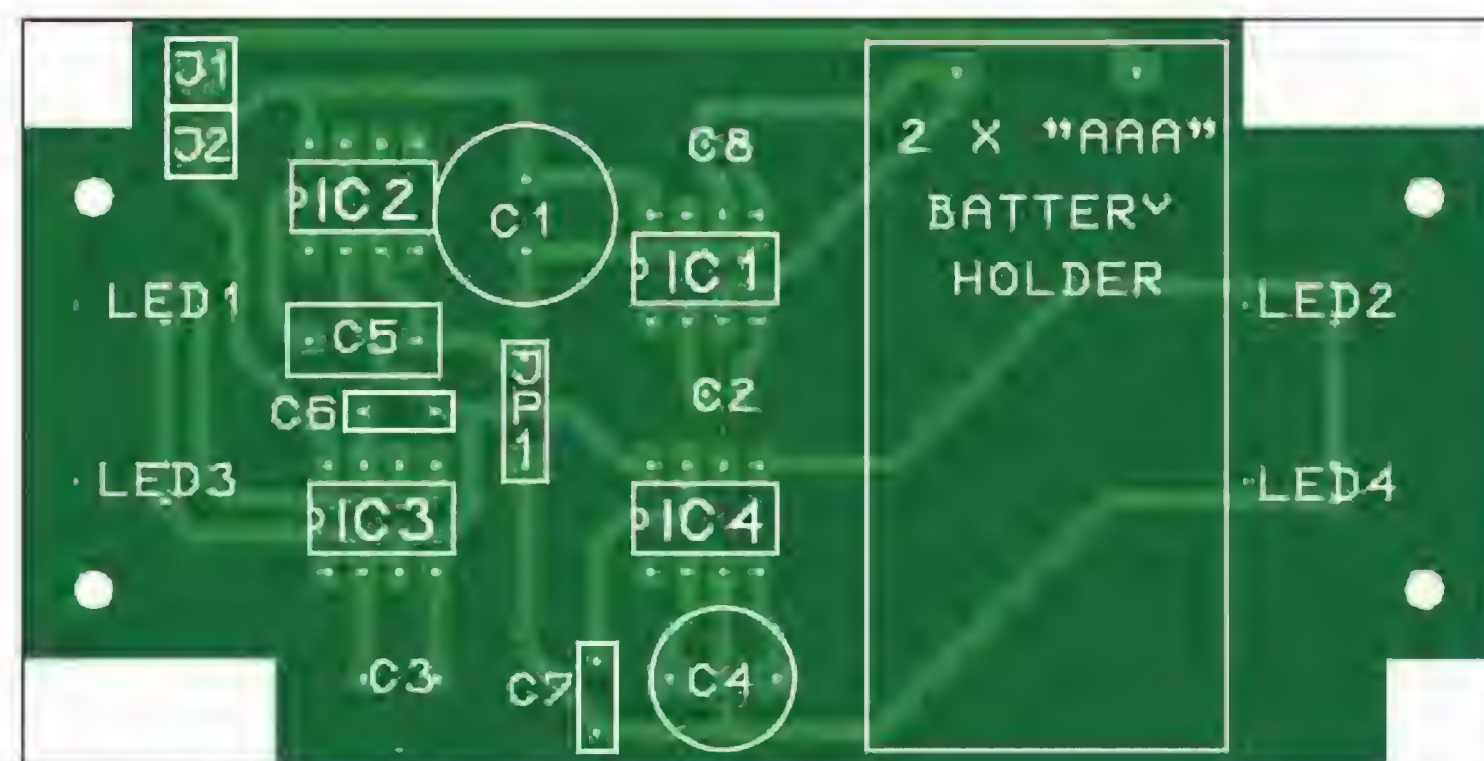

Figure 2. Parts placement of the proposed PCB, top view of board and trace pattern of board, viewed through top of board. (author's board design)

Internet Links and References

[1] www.maxim-ic.com/datasheet/index.mvp/id/1017/t/al

[2] www.elektor.com/110216

The Chaos Machine
Analogue Computing Rediscovered (1)

By Maarten H. P. Ambaum and
R. Giles Harrison
(Department of Meteorology,
University of Reading, UK)

Analogue computers provide actual rather than virtual representations of model systems. They are powerful and engaging computing machines that are cheap and simple to build. This two-part Retronics article helps you build (and understand!) your own analogue computer to simulate the Lorenz butterfly that's become iconic for Chaos theory. First, however, some history and background.
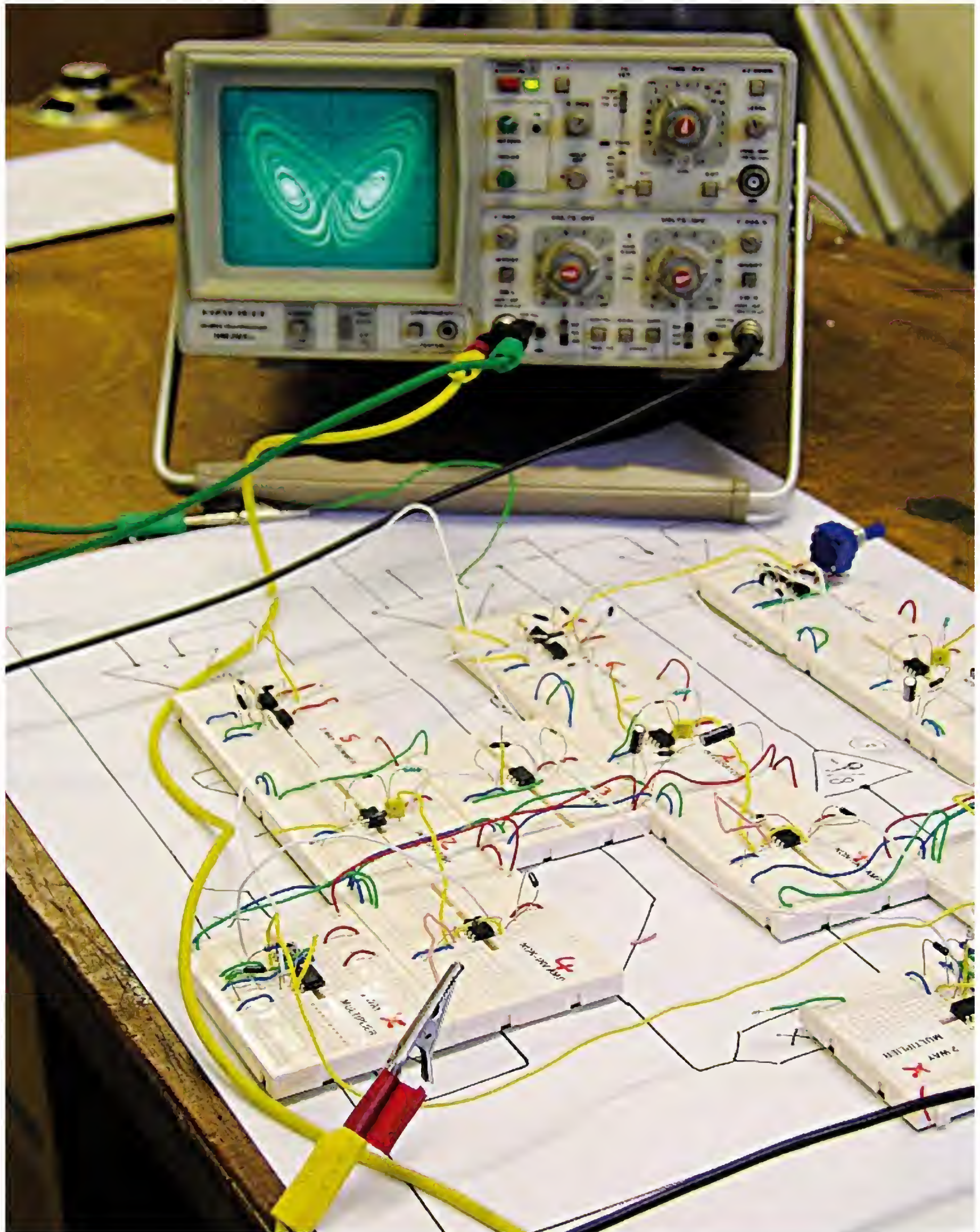
For some of us it may be surprising that, before the mid sixties, hardly any computing in real-time applications was done by a digital computer. Instead, analogue computers were used because of their speed and relative reliability. Analogue computers are machines that are built to behave as the system we want to compute.
A famous example is the Phillips MONIAC computer from the 1950s [1] (**Figure 1**) which used water flow through Perspex pipes to model the flow of money in an economy. However, in most practical applications, electronic analogies were used. The word analogue refers to the behaviour of the computer being analogous to that of the

system we want to simulate. In contrast, the word digital refers to the process of transforming the behaviour of a system to a stream of numbers or digits calculated by a numerical algorithm. Although this is the origin of the word analogue, its meaning has now evolved to describe anything that is not digital.

Modern analogue computers
In the sixties it became clear that the digital computer would rapidly overtake the analogue computer. Advances in chip technology made the digital computer reliable and available to a large num-

ber of people and organisations. However, precisely because of the advances in chip technology driven by the digital revolution, we can now build very cheap and very accurate analogue computers as well.

As part of an art-science collaboration in our Department, we decided to exploit the accuracy of modern analogue electronics in an exhibit of an analogue computer for the *Lorenz model* which produces the butterfly that became the iconic cartoon for the science of chaos and the unpredictability of weather (See, for example, James Gleick's *Chaos: Making a new science* for a wonderful introduction to chaos theory and its history).

Our building of the analogue computer turned out to be an inspiring and illuminating experience. Here we discuss some of the remarkable properties of analogue computers, perhaps no longer widely appreciated. In next month's instalment we will also describe how to make the analogue computer that simulates the Lorenz model. We call it the *Chaos Machine*.

Butterflies; poltergeists; mathematics

The Lorenz equations were developed in 1963 by the meteorologist Ed Lorenz to mimic the flow of air heated from below [2]. They are a set of three coupled equations that describe the time evolution of three variables X, Y, and Z,

$$dX/dt = \sigma(Y-X)$$

$$dY/dt = \rho X - Y - XZ$$

$$dZ/dt = XY - \beta Z$$

The link of these three equations to actual air flow is rather obscure, and they do not work very well anyway.

What Lorenz did discover was that when he chose the three tuneable parameters (σ; ρ; β) in his model carefully it would behave in an erratic and unpredictable way: chaos. This was completely unexpected and paved the way to a revolution in science. If the three variables X, Y, and Z are plotted as a moving point in three-dimensional space, we get the famous Lorenz butterfly, a fractal floating in three-dimensional space, see **Figure 2**.

In our Chaos Machine we can feed two of the voltages that represent the X, Y, and Z, to an oscilloscope in XY-display mode to see it draw an electronic version of the butterfly. We can tune the three variables to produce various shapes of the butterfly. We can also feed the channels to an audio amplifier to hear the sound of chaos. This turns out to be a remarkably unsettling experience: the *Chaos Machine* screeches and screams in the most bizarre ways with the soul of an electronic poltergeist.

How do analogue computers work?

An electronic analogue computer solves equations by representing values of variables by voltages in a circuit. Wires connect modules that perform specific arithmetic operations. For example, a subtraction module will have two input connectors and one output con-
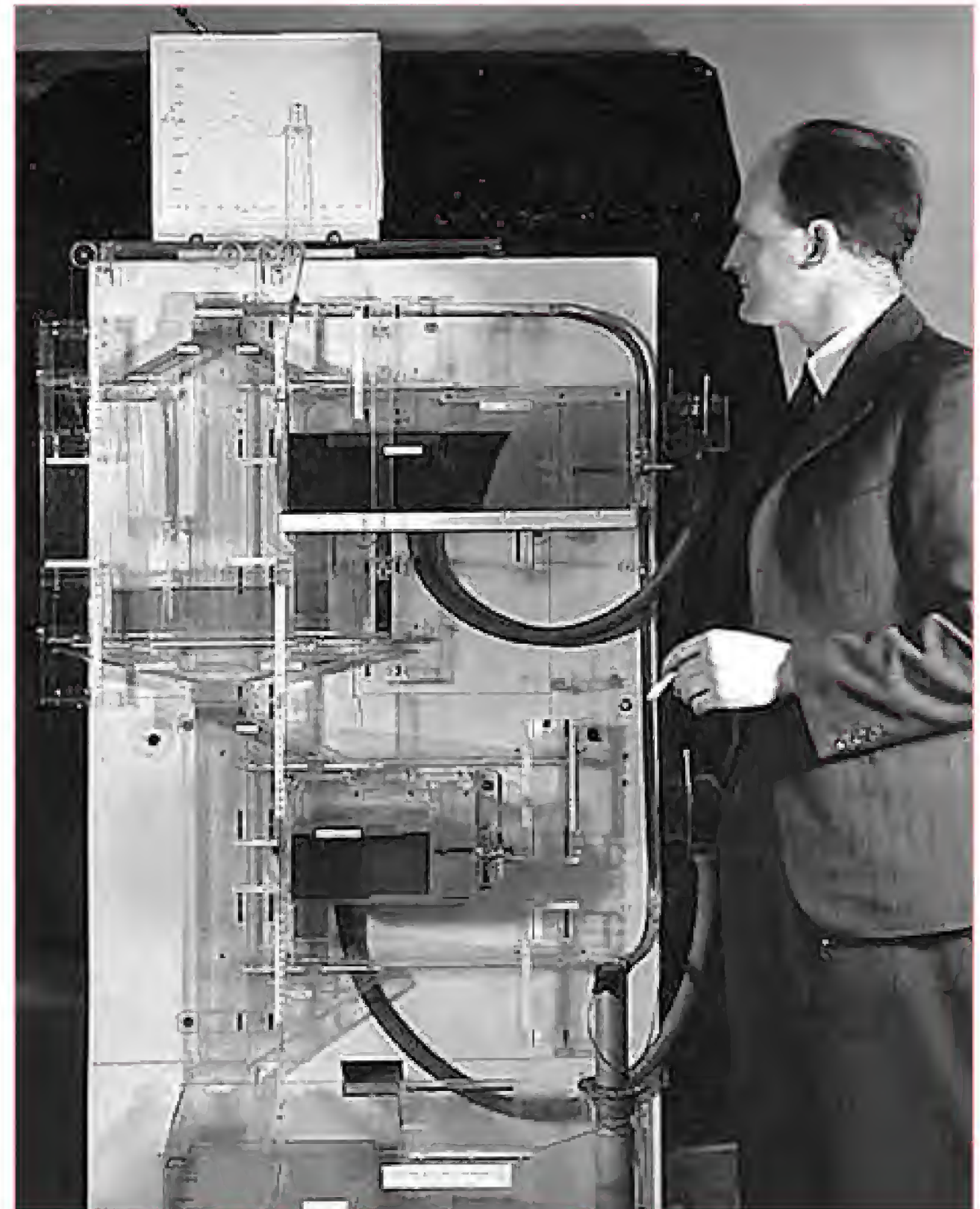

Figure 1. Professor A.W.H (Bill) Phillips was an LSE economist known for the 'Phillips curve' and he developed MONIAC, an analogue computer that modelled economic theory with water flows. *Image:* Wikimedia Commons.
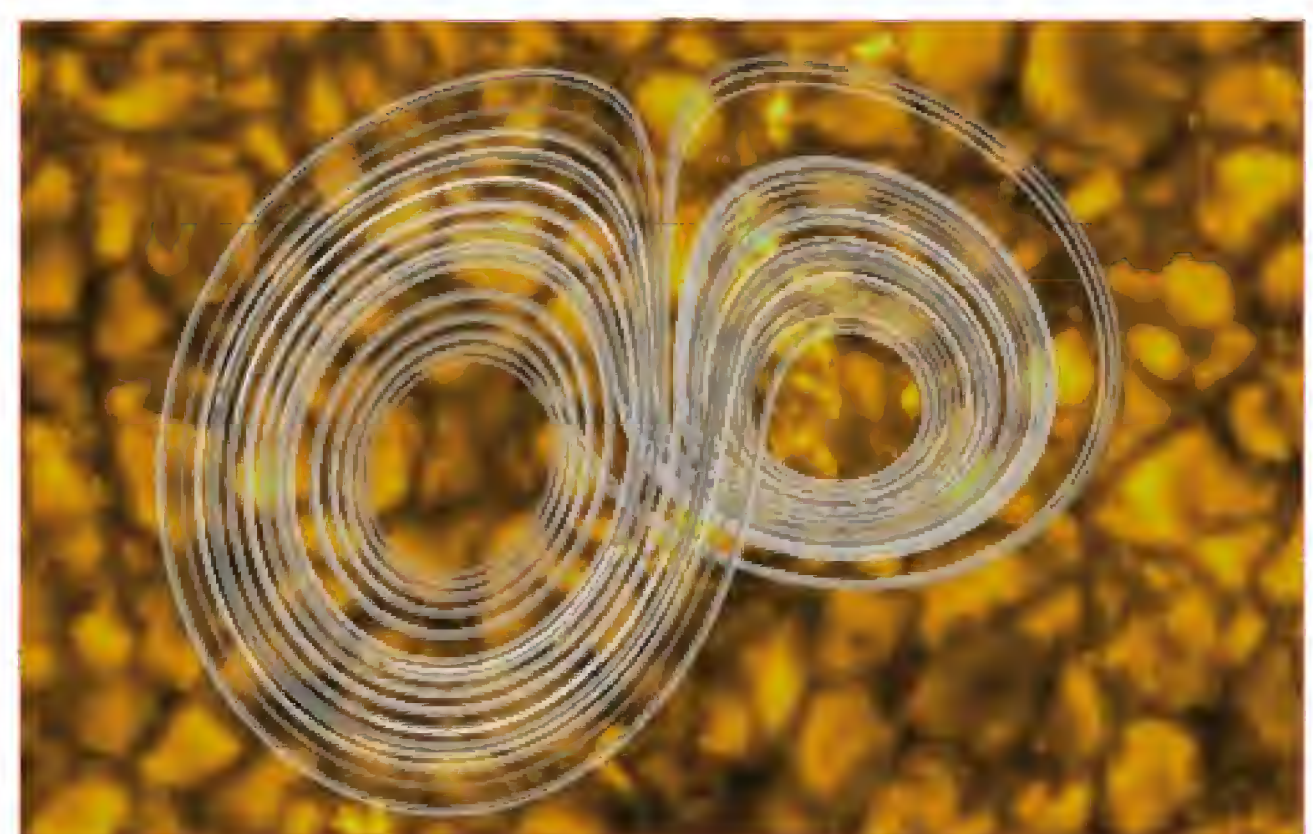

Figure 2. The Lorenz butterfly; the background is an image of solar convection, the original inspiration of the Lorenz equations.

nector where the output voltage equals the difference between the input voltages. This particular module is in fact simply a differential amplifier with unit gain.

The topology of an analogue computer is similar to that of our brain with the axons being represented by the wires, the cell body by the arithmetic modules, and the input ports by the dendrites. Contrast this with a digital computer. In a digital computer variables are stored in memory spaces which are then occasionally operated upon by copying these memory spaces to the central processor which then changes the values of variables in other memory spaces. Digital computers only change values of variables if the central processor says this should happen, and if so, they alter successively. In an analogue computer values always remain consistent. So, if for three variables a, b, and c we have $a+b = c$ then in an analogue computer this will always be the case. There is no internal clock speed; calculations happen instantaneously. In a digital computer this is only valid after the central processor has performed this addition and then only until either a or b are updated again.

Time integration is also a very natural process for an analogue computer. The input and the output voltages of a time integrating module are always consistently related: at all times the output voltage is equal to the time-integral of the input voltage. There are no time steps involved, as would be the case for a numerical integration routine. Numerical instability of integration routines is not an issue, nor are computing or storage overheads. The basic circuit of an integration module is shown in **Figure 3**. Integration in time occurs by converting a voltage to a current through use of an operational amplifier and then using this current to charge a capacitor. The instantaneous voltage across the capacitor is the time-integrated value of the input voltage. The schematic shows an electronic circuit able to perform integration in time of a varying input voltage. It is based on two operational amplifiers A1 and A2 each having inverting (–) and non-inverting (+) inputs and an output terminal. A time varying voltage $V_1(t)$ is applied to A1, which drives the integrator circuit comprising R, C and A2. The output voltage $V_2(t)$ is (minus) the time integral of the input voltage, scaled by (1/RC). A1 is a unit gain buffer stage, included solely to prevent loading of the originating voltage source but permitting a wide choice of values for R. (The additional resistor with A2 is for compensation and does not form part of the functional circuit.) For clarity, the necessary power supplies are not shown.

Other arithmetic operations can also be performed with the help of operational amplifiers. For example, subtracting two voltages is achieved using a differential amplifier of unit gain. Multiplication

and other related operations are more complicated to implement, requiring many op amp stages.

Analogue computers do not require a memory to work. This makes them essentially equivalent to the systems we try to simulate. A swinging pendulum does not have a memory of its previous states. One could connect an analogue computer to analogue–to–digital converters if digital storage or exact measurements are required. This construction can also be used to build a hybrid analogue–digital computer. A purist who wants to stay away from any digital technique can use a tape recorder or chart recorder for storage, also circumventing the difficulties of aliasing which arise in a sampled system.

Analogue computers are relatively hard to program: programming the computer is the same as building the computer. Clearly this flexibility is where digital computers are far superior. Also, in a digital computer it is easy to allocate memory spaces to store a set of variables, while in an analogue computer each variable is associated with a separate signal wire. Although a digital computer requires much more complex hardware for variable storage, it can use the same hardware configuration to tackle different virtual problems. A digital computer is a universal Turing machine, that is, a machine that can be used to solve different problems; an analogue computer can only solve one single problem.

Another fundamental difference between analogue and digital computers is that a digital computer calculates an approximated virtual representation of the model system, whereas an analogue computer is an actual electronic copy of the system. If we want to simulate a swinging pendulum with an analogue computer, we build an electronic system that oscillates exactly like the swinging pendulum. The computer becomes an electronic version of the swinging pendulum itself. This is a very appealing property of analogue computers. Think of the Lorenz system that we use in our *Chaos Machine*. Apart from a very artificial set-up, there is no actual physical representation of the system; it was designed as a mathematical system. Analogue computers are the only way we can get genuine physical representations of such mathematical systems.

Figure 3. Basic circuit of the time integration module.

How fast are they?

People who see an analogue computer for the first time often ask: how fast is it compared to a modern digital computer? In fact, their speeds are hard to compare. In a digital computer speed is limited by the clock speed of the processor and the speed at which variables can be loaded into and out of the processor. One such calcu-
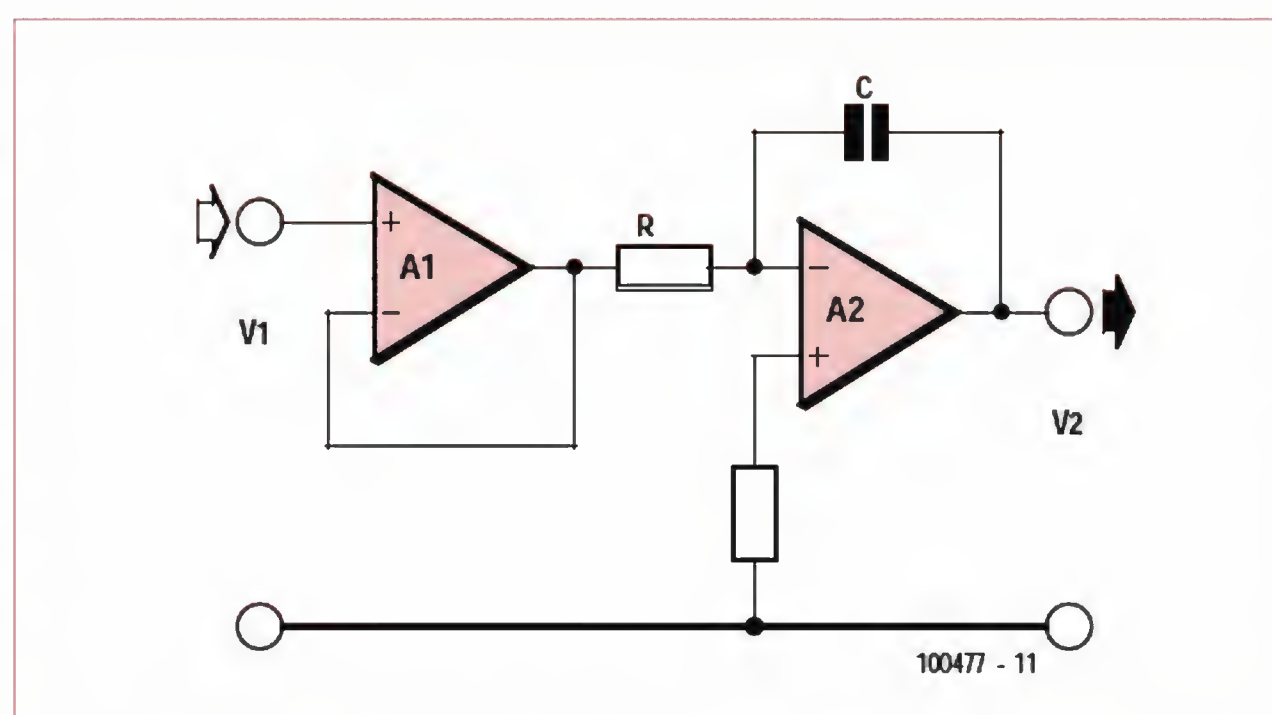
lation may typically take a nanosecond or so (one thousand millionth of a second). In an analogue computer the speed is limited by the speed at which the operational amplifiers, the key building blocks of analogue computers, can follow changes in input voltages (the *slew rate*). Operational amplifiers can change over time scales of a few nanoseconds and for most practical purposes this does not limit the computer's operation. However, analogue computers do not perform calculations as such; they perform simulations. Asking how fast an analogue computer calculates is the same as asking how fast the swinging pendulum calculates its motion.

Nevertheless, the 'speed' comparison can be made more precise. An analogue time integration module has an intrinsic timescale set by $R{\times}C$, the resistance and capacitance, respectively, of two components in the module. In other words, speed in a digital computer is limited by its clock speed, while 'speed' in an analogue computer can be arbitrarily defined by choosing different components. There is a practical limit to which we can increase the speed of the analogue computer set by the finite slew rate of the operational amplifiers and the stray capacitances in the system, both of which act to damp away the very highest frequencies.

Fortunately, dedicated analogue function chips are cheaply available which contain optimized log and antilog converters, providing multiplication, divisions, and square roots with excellent accuracy and temperature stability — and truly enormous speed.

Operational amplifiers

Advances in electronic components have had major benefits for many scientific activities, but they also now make the implementation of analogue computing straightforward. The key building block of an electronic analogue computer is the operational amplifier, a general purpose electronic device which can be configured to perform the different mathematical operations (integration, addition, multiplication, scaling) required. General purpose amplifiers originated in the 1940s from military applications, particularly in anti-aircraft gunnery (although a mechanical analogue computer was used as recent as the Vietnam war in the 'Norden' bombsight to target bombs dropped from aircraft). The description *operational amplifier* (or *op amp*), appeared in 1947, and the first commercial op amp — type K2-W — was produced by Philbrick in 1953, based on two

dual triode valves, see **Figure 4** and [3,4]. Solid state op amps followed in the 1960s, with the first integrated circuit op amp in 1965 (Jung's *Op Amp Applications Handbook* provides a good overview of the history and use of op amps).

Comparison between early and modern op amps illustrates how the steady improvements have made analogue computing ever more practical. The thermionic K2-W had a specified drift of ±5 mV per day, whereas the integrated circuit OP97, used in our Lorenz design, has drift dominated by thermal changes, at 0.6 µV/°C. The power requirements are also dramatically different. A K2-W required power supplies of ±300 V and 6.3 V, at about 10 mA and 0.6 A, whereas the OP97 requires ±15 V at 0.6 mA. Early analogue computers therefore had a substantial physical volume associated with each computing stage, as well as power dissipation and heat generation.

Integrated circuit analogue computers are now compact, and drift is no longer a characteristic feature. Individual op amp stages are also relatively cheap, allowing complex systems to be readily simulated. A benefit of their low cost is that extra stages can easily be included, which although not essential to the computing function, may relax constraints on the components required.

The final analogue computer is an assembly of independent circuit modules, combined to solve one specific problem, but reusable for other applications. With such a modular approach the 'programming' of the computer is a fairly simple job which requires hardly any knowledge of the electronics involved.

Next month's second and final instalment discusses the elements that go into building the *Chaos Machine*.

(100477)

Figure 4. The Philbrick K2-W is generally considered the first commercial operational amplifier.

Internet Links and References

[1] http://en.wikipedia.org/wiki/MONIAC_Computer

[2] http://mathworld.wolfram.com/LorenzAttractor.html

[3] www.philbrickarchive.org/

[4] Philbrick K2-W, the mother of all op amps,
     Elektor (Retronics) October 2009.

Light Sensor
with Twilight Detection
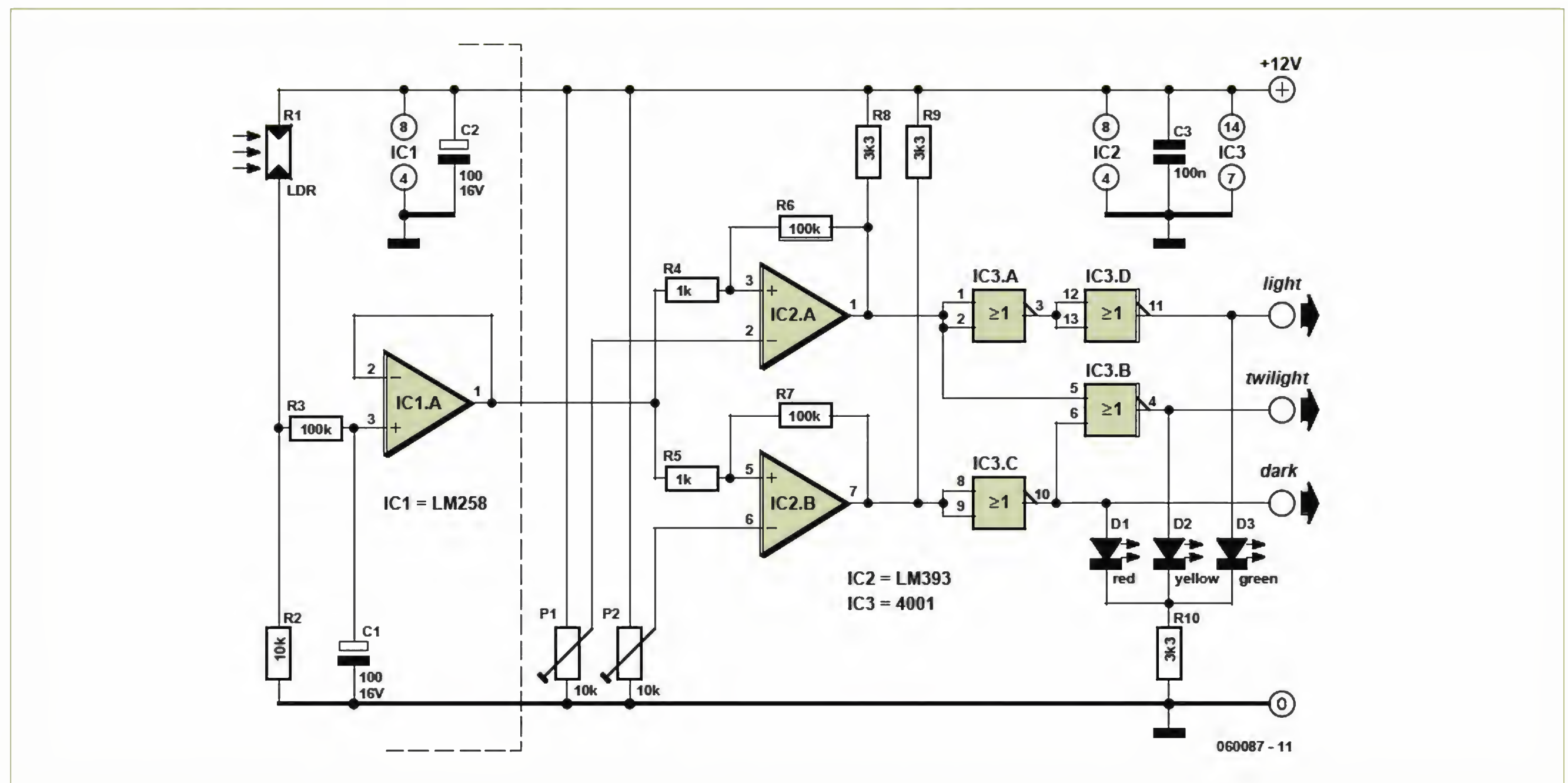
By Heino Peters (The Netherlands)

This is not the first light sensitive circuit to be published in Elektor magazine. This circuit however, distinguishes itself that in addition to light and dark it can also signal twilight (dusk). This lets you automatically turn on a light in the living room when it becomes dark and turn on a lamp in a dark hallway when dusk sets in. The circuit described here generates a logic signal on three separate outputs for light, twilight and dark. The transition thresholds are set with two trimpots.

The part of the circuit that is to the left of the dashed line can be located outside, on the roof, for example. This is possible because the LM258 can withstand frost, unlike the LM358, for instance. R1 and R2 together form a light dependent voltage divider, the voltage variations of which are damped by R3 and C1. This is desirable so that the circuit is less sensitive to birds that could cause the curtains to be closed when they fly across the sensor.

Opamp IC1a is wired as a buffer, so that the voltage that is seen by the remainder of the circuit does not deviate too much from the voltage 'on the roof'. Any arbitrary LDR is suitable for R1, but do make sure that the

voltage level at pin 3 of IC1a is at least 2 V below the power supply voltage when it is light. This is because that is the maximum voltage that IC1 and IC2 can tolerate at their inputs. Otherwise fit an additional resistor of, for example, 2.2 kΩ between R1 and the power supply.

Two comparators (IC2a and IC2b) compare the incoming voltage with the threshold voltages set by P1 and P2. R4 and R6 (R5 and R7) prevent that that output of IC2a (IC2b) will jitter around the threshold. R8 and R9 have been added because IC2 has open-collector outputs.

It is actually already possible to determine whether it is light, dark or twilight by looking at the outputs of IC2a and IC2b, but the four gates of IC3 turn these into three separate signals. To make the adjustment easier, there are three LEDs of different colour connected to the outputs: green for light, yellow for twilight and red for dark. In the box is a description of the steps that are necessary to adjust the circuit. It is best to do this towards the evening, that is when it is still light outside before the fall of dusk.

To adjust the threshold values, P1 is intended for the transition from light to twilight and P2 for the transition from twi-

light to dark. With a correctly adjusted circuit, the voltage at the wiper of P1 has to be lower than the voltage at the wiper of P2. Because the outputs of the CMOS gates cannot drive heavy loads, low-current LEDs are essential. These have enough with only 2 mA, while ordinary LEDs will often need 20 mA. The power supply voltage can be from 9 VDC to 15 VDC.

(060087-1)

Adjustment

1. First turn the wipers of both P1 and P2 to ground. If all is well only the green LED should be on.

2. Wait until dusk falls.

3. Now turn P1 just to the point where the green LED turns off and the yellow LED just turns on.

4. Now wait until it is dark.

5. Turn P2 just to the point where the yellow LED turns off and the red LED turns on. The adjustment is now complete.

Hexadoku
Puzzle with an electronics touch

With the summer holidays ended for the most part we can safely return to the monthly dose of Hexadoku to keep you busy for a couple of hours. After the monster puzzle in the July & August edition it's back to the grind with a regular 16 x 16 grid Hexadoku. Enter the right numbers in the puzzle. Next, send the ones in the grey boxes to us and you automatically enter the prize draw for one of four Elektor Shop vouchers. Have fun!

The instructions for this puzzle are straightforward. Fully geared to electronics fans and programmers, the Hexadoku puzzle employs the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation. Correct entries received enter a draw for a main prize and three lesser prizes. All you need to do is send us the numbers in the grey boxes.

Solve Hexadoku and win!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for one Elektor Shop voucher worth £ 80.00 and three Elektor Shop Vouchers worth £ 40.00 each, which should encourage all Elektor readers to participate.

Participate!

Before October 1, 2011, send your solution (the numbers in the grey boxes) by email, fax or post to
Elektor Hexadoku – 1000, Great West Road – Brentford TW8 9HH United Kingdom.
Fax (+44) 208 2614447   Email: hexadoku@elektor.com

Prize winners
The solution of the June 2011 Hexadoku is: B18AD.
The Elektor £80.00 voucher has been awarded to Mads Thorup (Denmark).
The Elektor £40.00 vouchers have been awarded to Wolfgang Kallauch (Germany),
Tommy Vanhullebusch (Belgium) and Timon Zijnge (Netherlands).
Congratulations everyone!

	9						4			B	D	2			
	A				D	9	4			1	2	E			
7					B		5			A				C	
	8			5			1	3	D	6		F	4		A
E		B				D				4	1	7	0		
0			D			F		1	7			8	B	2	
			2									A	D	3	
6	3				1		B		2		5		E		
		E		7		3		D		4				2	B
2	B	0									8				
9	7	1				8	6		E			4			D
	C	3	F	E				A				5			6
3		4	B		E	A	7	0				9		8	
	E			D			F		B						9
		1	0	B			E	F	8				4		
		8	C	1		9							5		

5	6	D	4	B	7	2	8	A	E	1	0	3	C	F	9
2	E	F	0	4	C	9	A	B	6	3	8	5	7	D	1
7	3	1	A	D	E	5	F	2	4	9	C	B	0	6	8
C	8	9	B	0	1	3	6	5	D	7	F	E	A	2	4
6	1	C	9	5	3	F	B	7	8	0	4	D	2	E	A
D	B	E	F	6	2	7	4	9	1	5	A	C	8	0	3
0	4	2	7	8	A	D	E	C	F	6	3	9	5	1	B
3	5	A	8	C	0	1	9	D	2	B	E	4	F	7	6
4	7	0	C	E	9	B	1	8	A	D	2	6	3	5	F
8	9	5	6	2	D	4	0	E	3	F	7	1	B	A	C
E	D	B	1	F	8	A	3	0	5	C	6	2	9	4	7
F	A	3	2	7	5	6	C	1	B	4	9	0	D	8	2
9	C	6	E	1	B	8	D	F	0	A	5	7	4	3	2
A	0	8	5	3	6	C	2	4	7	E	B	F	1	9	D
1	2	4	3	9	F	0	7	6	C	8	D	A	E	B	5
B	F	7	D	A	4	E	5	3	9	2	1	8	6	C	0

Going Strong

A world of electronics from a single shop!

NEW!

Controller Area Network Projects

Dogan Ibrahim
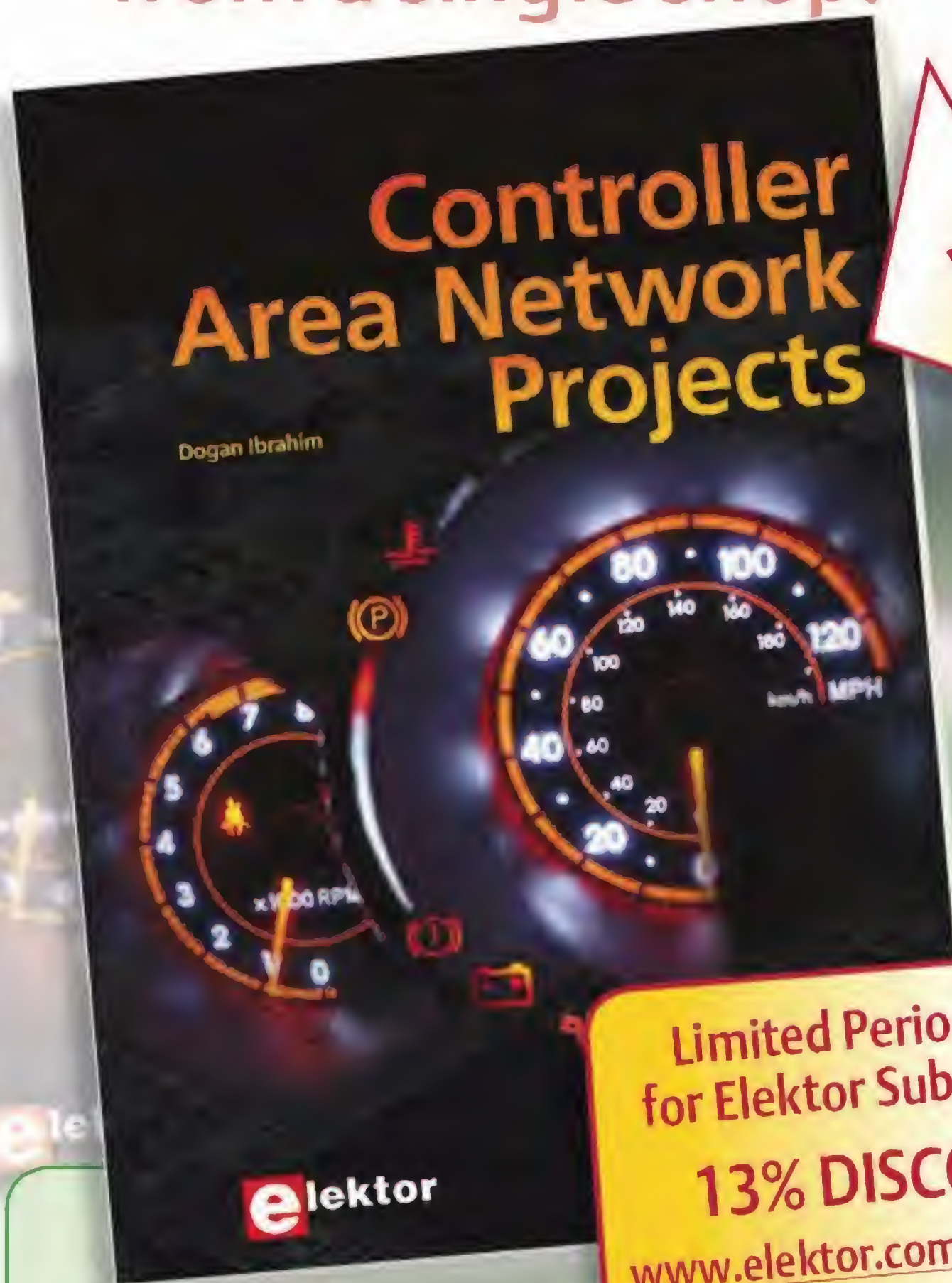
elektor

Limited Period Offer for Elektor Subscribers!
13% DISCOUNT
www.elektor.com/september

Free mikroC compiler CD-ROM included

Controller Area Network Projects

This book is written for students, for practising engineers, for hobbyists, and for everyone else who may be interested to learn more about the CAN bus and its applications. The aim of the book is to teach you the basic principles of CAN networks and in addition the development of microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters. This book comes complete with a free CD containing a special demo version of the mikroC compiler which supports the key microcontrollers including: PIC, dsPIC, PIC24, PIC32 and AVR. This special version additionally features an advanced CAN library of intuitive and simple-to-use functions to encourage programming with easy and comfortable development of CAN networks.

260 pages • ISBN 978-1-907920-04-2 • £29.50 • US $47.60

Bestseller!

LabWorX: Straight from the Lab to your Brain

Mastering the I²C Bus

Mastering the I²C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I²C Bus and its applications. Besides th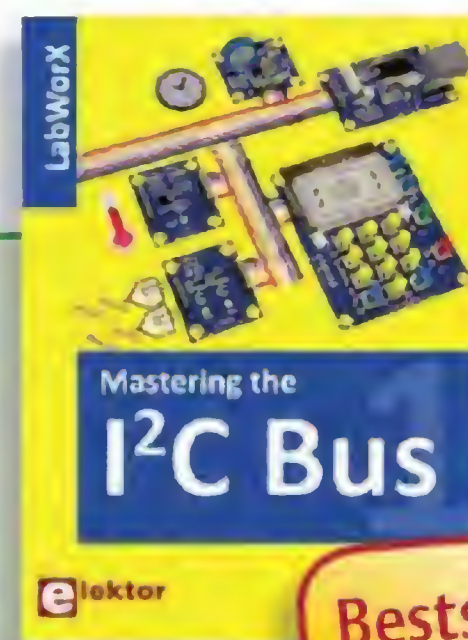e Bus protocol plenty of attention is given to the practical applications and designing a solid system. The most common I²C compatible chip classes are covered in detail. Two experimentation boards are available that allow for rapid prototype development. These are completed by a USB to I²C probe and a software framework to control I²C devices from your computer.

248 pages • ISBN 978-0-905705-98-9
£29.50 • US $47.60
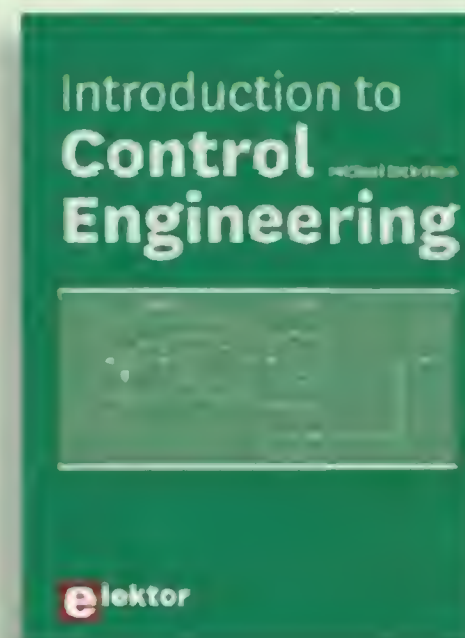
Solutions for control system applications

Introduction to Control Engineering

This book is intended as a source of reference for hardware and software associated with instrumentation and control engineering. Examples are presented from a range of industries and applications. Throughout the book, circuit diagrams and software listings are described, typical of many measurement and control applications. The hardware and software designs may be used as a basis for application by the reader. The book contains examples of PIC, PLC, PAC and PC programming.

164 pages • ISBN 978-0-905705-99-6
£27.50 • US $44.40
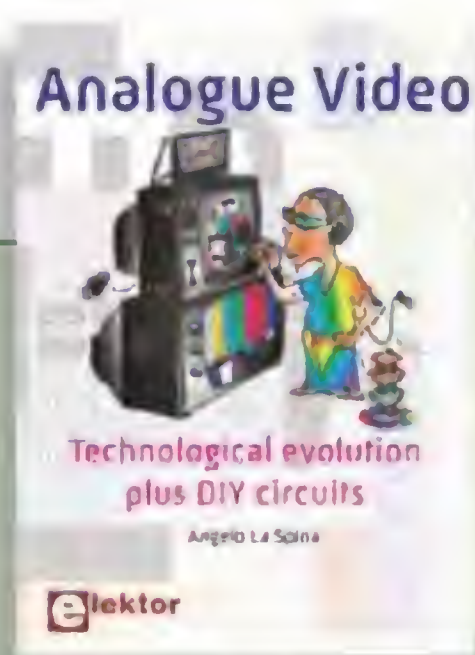
Analogue Video

Technological evolution plus DIY circuits

Analogue Video

This book is intended for electronics enthusiasts and professionals alike, who want a much deeper understanding of the incredible technology conquests over the pre-digital decades that created video. It details evolution of analogue video electronics and technology from the first electro-mechanical television, through advancements in Cathode Ray Tubes, transistor circuits and signal processing, up to the latest analogue, colour-rich TV, entertainment devices and calibration equipment.

222 pages • ISBN 978-0-905705-96-5
£26.50 • US $42.80

A highly-practical guide

Linux – PC-based Measurement Electronics

If you want to learn how to quickly build Linux-based applications able to collect, process and display data on a PC from various analog and digital sensors, how to control circuitry attached to a computer, then even how to pass data via a network or control your embedded system wirelessly and more – then this is the book for you! The book covers both hardware and software aspects of designing typical embedded systems using schematics, code listings and full descriptions.

264 pages • ISBN 978-1-907920-03-5
£29.50 • US $47.60

Enhanced second edition: 180 new pages

Design your own Embedded Linux Control Centre on a PC

The main system described in this book re-uses an old PC, a wireless mains outlet with three switches and one controller, and a USB webcam. All this is linked together by Linux. This book will serve up the basics of setting up a Linux environment – including a software development environment – so it can be used as a control centre. The book will also guide you through the necessary setup and configuration of a webserver, which will be the interface to your very own home control centre. New edition enhancements include details of extending the capabilities of your control center with ports for a mobile phone (for SMS messaging) and the Elektor "thermo snake" for low-cost networked real-time thermal monitoring of your house and outbuildings. Now you can additionally also send all kinds of useful temperature and sensor warnings to a mobile phone. All software needed will be available at the Elektor website.

416 pages • ISBN 978-1-907920-02-8
£34.50 • US $55.70

Circuits, ideas, tips and tricks from Elektor

CD 1001 Circuits

This CD-ROM contains more than 1000 circuits, ideas, tips and tricks from the Summer Curcuits issues 2001-2010 of Elektor, supplemented with various other small projects, including all circuit diagrams, descriptions, component lists and full-sized layouts. The articles are grouped alphabetically in nine different sections: audio & video, computer & microcontroller, hobby & modelling, home & garden, high frequency, power supply, robotics, test & measurement and of course a section miscellaneous for everything that didn't fit in one of the other sections. Texts and component lists may be searched with the search function of Adobe Reader.

ISBN 978-1-907920-06-6
£34.50 • US $55.70

More than 70,000 components

CD Elektor's Components Database 6

This CD-ROM gives you easy access to design data for over 7,800 ICs, more than 35,600 transistors, FETs, thyristors and triacs, just under 25,000 diodes and 1,800 optocouplers. The program package consists of eight databanks covering ICs, transistors, diodes and optocouplers. A further eleven applications cover the calculation of, for example, zener diode series resistors, voltage regulators, voltage dividers and AMV's. A colour band decoder is included for determining resistor and inductor values. All databank applications are fully interactive, allowing the user to add, edit and complete component data.

ISBN 978-90-5381-258-7
£24.90 • US $40.20

CD/DVD-ROMs

CD/DVD-ROMs

More than 25 projects based on the Elektor ATM18 board

CD ATM18 Collection

This CD-ROM contains all articles from the popular ATM18-CC2 series published in Elektor magazine. From RFID Reader and Bluetooth linking right up to a chess computer! Project software and PCB layouts in PDF format are included. What's more, this CD also contains a Bascom AVR programming course and helpful supplementary documentation.

ISBN 978-0-905705-92-7
£24.50 • US $39.60

All articles in Elektor Volume 2010

DVD Elektor 2010

This DVD-ROM contains all editorial articles published in Volume 2010 of the English, Spanish, Dutch, French and German editions of Elektor. Using the supplied Adobe Reader program, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article. With this DVD you can also produce hard copy of PCB layouts at printer resolution, adapt PCB layouts using your favourite graphics program, zoom in / out on selected PCB areas and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-267-9
£23.50 • US $37.90

Wireless OBD-II

(April 2011)

The cheapest way to diagnose faults on a modern car is to connect its OBD-II interface to a (notebook) PC running suitable diagnostics software. However, a wired connection is not always the most suitable, and selfcontained OBD testers are a rather expensive and less flexible alternative to using a PC. An interesting option is a wireless OBD interface with a radio interface to a PC: this homebrew solution allows the choice of using either Bluetooth or ZigBee.

OBD2- Zigbee or Bluetooth interface kit with all parts and enclosure

Art.# 100872-71
£111.00 • US $179.10 (Zigbee)
Art.# 100872-72
£111.00 • US $179.10 (Bluetooth)

Bestseller!

Pico C Meter

(April 2011)

RF and radio repair fans probably do need to be told, but when it comes to measurements below 200 pF or so, modern DMMs will produce coarse if not ridiculous results. Elektor's purpose-designed Pico C does a far better job. Beating many DMMs hands down, this little instrument easily and accurately measures capacitances down to fractions of a picofarad!

Kit of parts incl. Elektor Project Case, programmed microcontroller, LCD and PCB

Art.# 100823-71 • £73.40 • US $118.40

SatFinder

(March 2011)

Those of you who regularly need to realign a satellite TV dish will find this gadget extremely valuable. Caravan owners and campers on long journeys who crave their home TV channels can now keep up with developments in sports, news and the soaps back home with the help of the SatFinder. This GPS based design includes a database containing positional information of a number of popular TV satellites. With the help of GPS data it calculates the precise angles to find the satellite first time!

Kit of parts including Controller, display and PCB (European Version)

Art.# 100699-71 • £71.20 • US $114.90
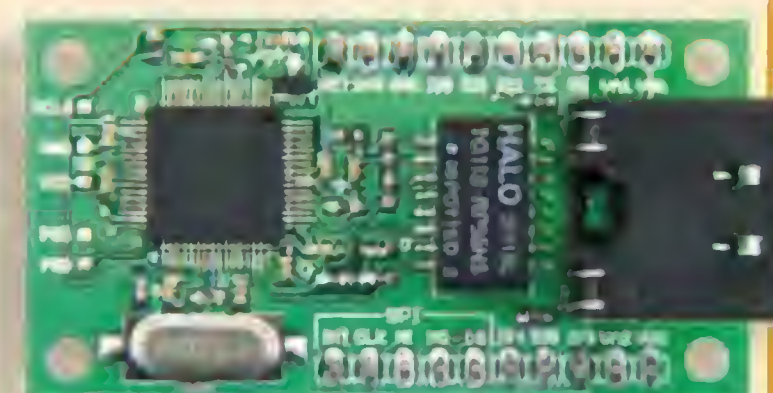
NetWorker

(December 2010)

An Internet connection would be a valuable addition to many projects, but often designers are put off by the complexities involved. The 'NetWorker', which consists of a small printed circuit board, a free software library and a ready-to-use microcontroller-based web server, solves these problems and allows beginners to add Internet connectivity to their projects. More experienced users will benefit from features such as SPI communications, power over Ethernet (PoE) and more.

Module, ready assembled and tested

Art.# 100552-91 • £53.00 • US $85.50

Kits & Modules

82   Prices and item descriptions subject to change. E. & O.E

09-2011   elektor

September 2011 (No. 417)   £   US $

+ + + Product Shortlist September: See www.elektor.com + + +

July/August 2011 (No. 415/416)

2/4/6-hour Timer

110219-41 .... Programmed controller PIC12F675 DIL8................... 8.85.......14.30

Morse Clock

110170-41 .... Programmed controller ATTiny4520-PU dip 8............ 8.85.......14.30

Return of the Elex Prototyping Board

ELEX-1 .... Printed circuit board Universal exp. board.................. 4.90.........7.90

ELEX-2 .... Printed circuit board Double version universal
                        exp. board.. 8.85.......14.25

ELEX-4 .... Printed circuit board Quadruple
                        version universal exp. board .... 16.00.......25.80

Arc Welding Effect for Model Railway Layouts

110085-41 .... Programmed controller PIC10F200-I/P (DIP8) .... 8.85.......14.30

Slave Flash for Underwater Camera

100584-41 .... Programmed controller PIC12F675i/p DIL8 .... 8.85.......14.30

WAV Doorbell

110080-41 .... Programmed controller ATmega328p DIL28 .... 11.55.......18.60

RGB Solar Lamp

100581-41 .... Programmed controller ATTiny13 DIL8.................... 8.85.......14.30

Jogging Timer

110160-41 .... Programmed controller ATTiny44-20PU dip 14 .... 8.85.......14.30

Battery Charge Monitor

110154-41 .... Programmed controller PIC16F873A...................... 12.50.......20.10

Roadwork Traffic Signals for Modellers

110203-41 .... Programmed controller ATTiny13 (8-pen DIP) .... 8.8.......14.30

Make Your R8C/13 Speak CAN

050179-91 .... 16-bit R8C microcontroller board .... 10.90.......17.60

June 2011 (No. 414)

Elektor OSPV1

110320-91 .... Kit... 975.00...1570.00

Geolocation with the ATM18

071035-91 .... ATM18 Controller module, partly populated............... 9.50.......15.40

071035-92 .... ATM18 Testboard, partly populated......................... 29.90.......48.30

071035-93 .... LCD board, SMD populated incl. pinheaders............. 23.00.......37.10

Here comes the Bus! (6)

110258-91 .... USB/RS485 Converter, ready made module .... 22.20.......35.70

E-blocks: Flowcode RC5

EB007 .... E-block Switch Bboard... 14.40.......23.90

EB058 .... E-block Color Graphics Display............................... 66.10.....109.60

EB060 .... E-block RC5 Infrared Board .... 30.00.......49.80

EB064 .... E-block dsPIC/PIC24 Multiprogrammer .... 96.00.....159.40

May 2011 (No. 413)

Microphone Conferencing System

100465-1 ...... Printed circuit board.. 8.85.......14.30

Elektor Proton Robot

110263-71 .... Kit of parts All Inclusive
                        (Body + Head + Audio + Gripper + PIC Add-on)....... 1085.00...1745.00

110263-72 .... Kit of parts All Inclusive
                        (Body + Head + Audio + Gripper + AVR Add-on) ..... 1085.00...1745.00

110263-78 .... Ready assembled and tested, PIC Add-on.................. 34.00.......55.00

110263-79 .... Ready assembled and tested, AVR Add-on .... 34.00...... 55.00

110263-91 .... Ready assembled and tested, PIC version............. 1475.00...2375.00

110263-92 .... Ready assembled and tested, AVR version............. 1475.00...2375.00

1-Channel DMX512 Light Dimmer

EB006 .... E-block PIC MultiProgrammer .... 75.00.....121.00

TEFLCST4....... E-block Flowcode 4 for PICmicro .... 45.90.......74.10

Mobile, Text, CallerID

071035-72 .... Relay board with all parts and relays......................... 36.90.......73.80

071035-91 .... PCB, partly populated, ATM18 Controller module....... 9.50.......15.40

071035-92 .... PCB, partly populated ATM18 Test board .... 29.90.......48.30

071035-93 .... LCD board, SMD populated incl. pinheaders............. 23.00.......37.10

071035-95 .... Port extension board, SMD populated .... 13.40.......26.80

Retronics: The Chaos Generator (2)

After reading this month's Retronics instalment on a chaos generator, the editors (in chaotic fashion) got the idea to put the theory to the test using a handful of opamps soldered on to small boards. It all worked wonderfully well and never before did we see such strange shapes on an oscilloscope screen. From sea horses and butterflies right up to LSE's latest "business models". The audio signals too produced by the generator proved unnerving at times. You can read all about it in the October 2011 edition.

PCBino

In many electronics projects 'the PCB' plays second fiddle to the electronic design. Although it is hard to assemble a circuit without a circuit board, the design of the latter is often forgotten in favour of "how the thing works". To compensate this unjustness we decided to turn the tables. This article is all about the circuit board and the circuit on it is a trifle, really. Ladies and Gentlemen, your warm welcome please for....PCBino!

Dreams of flying electrically one day...

... have come true. The first approved-for-air transport aircraft with electric drive has been in production since 2004, while this year Airbus parent company EADS presented "Voltaire", their all-electric propulsion system concept for future commercial aircraft.
We report on the current status of the sadly neglected field of electric vehicles as well as on the Green Flight Challenge competition planned for September 25, 2011, which is supported by NASA with prize money amounting to 1.65 million dollars.

Article titles and magazine contents subject to change; please check the Magazine tab on www.elektor.com

Elektor UK/European October 2011 edition: on sale September 22, 2011.   *Elektor USA April 2011 edition: published September 19, 2011.*

w.elektor.com www.elektor.com www.elektor.com www.elektor.com www.elektor.com ww

Elektor on the web

All magazine articles back to volume 2000 are available online in pdf format. The article summary and parts list (if applicable) can be instantly viewed to help you positively identify an article. Article related items are also shown, including software downloads, circuit boards, programmed ICs and corrections and updates if applicable. Complete magazine issues may also be downloaded.

In the Elektor Shop you'll find all other products sold by the publishers, like CD-ROMs, DVDs, kits, modules, equipment, tools and books. A powerful search function allows you to search for items and references across the entire website.

Also on the Elektor website:
- Electronics news and Elektor announcements
- Readers Forum
- PCB, software and e-magazine downloads
- Time limited offers
- FAQ, Author Guidelines and Contact

Description	Price each	Qty.	Total	Order Code
CD 1001 Circuits **NEW**	£34.50			
Controller Area Network Projects **NEW**	£29.50			
LabWorx – Mastering the I²C Bus **NEW**	£29.50			
Linux – PC-based Measurement Electronics	£29.50			
Design your own Embedded Linux Control Centre on a PC	£34.50			
CD Elektor's Components Database 6	£24.90			
DVD Elektor 2010	£23.50			
	Sub-total			
	P&P			
	Total paid			

Prices and item descriptions subject to change.
The publishers reserve the right to change prices
without prior notification. Prices and item descriptions
shown here supersede those in previous issues. E. & O.E.

Name

Address + Post code

Tel.   Email

Date   –   –   Signature

EL09

METHOD OF PAYMENT
(see reverse before ticking as appropriate)

☐ **Bank transfer**

☐ **Cheque**
(UK-resident customers ONLY)

☐ **VISA**   ☐ **MasterCard** EUROCARD

Expiry date: _____

Verification code: _____

Please send this order form to*
(see reverse for conditions)

Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH
United Kingdom

Tel.: +44 20 8261 4509
Fax: +44 20 8261 4447
www.elektor.com
order@elektor.com

*USA and Canada residents should use $ prices,
and send the order form to:
Elektor US
PO Box 180
Vernon CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: order@elektor.com

Yes, I am taking out an annual subscription to Elektor and receive the E-book Microcontroller Basics totally free!*

(Please fill in your emailaddress below)

I would like:

☐ **Standard Subscription (11 issues)**

☐ **Subscription-Plus**
(11 issues plus the Elektor Volume 2011 DVD-ROM TIP
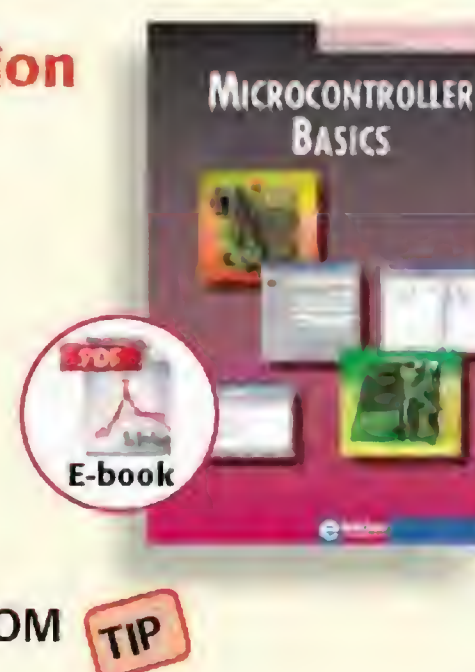+ exclusive access to www.elektor-plus.com)

* Offer available to Subscribers who have not held a subscription
to Elektor during the last 12 months. Offer subject to availability.
See reverse for rates and conditions.

Name

Address + Post code

Tel.   Email

Date   –   –   Signature

EL09

METHOD OF PAYMENT
(see reverse before ticking as appropriate)

☐ **Bank transfer**

☐ **Cheque**
(UK-resident customers ONLY)

☐ **VISA**   ☐ **MasterCard** EUROCARD

Expiry date: _____

Verification code: _____

Please send this order form to

Elektor
Regus Brentford
1000 Great West Road
Brentford TW8 9HH
United Kingdom

Tel.: +44 20 8261 4509
Fax: +44 20 8261 4447
www.elektor.com
service@elektor.com

ORDERING INSTRUCTIONS, P&P CHARGES

All orders, except for subscriptions (for which see below), must be sent BY POST or FAX to our Brentford address using the Order Form overleaf. Online ordering: www.elektor.com/shop
Readers in the USA and Canada should send orders, except for subscriptions (for which see below), to the USA address given on the order form. Please apply to Elektor US for applicable P&P charges. Please allow 4-6 weeks for delivery.
Orders placed on our Brentford office must include P&P charges (Priority or Standard) as follows: Europe: £6.00 (Standard) or £7.00 (Priority) Outside Europe: £9.00 (Standard) or £11.00 (Priority)

HOW TO PAY

All orders must be accompanied by the full payment, including postage and packing charges as stated above or advised by Customer Services staff.
Bank transfer into account no. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London. IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L. Currency: sterling (UKP).
Please ensure your full name and address gets communicated to us.
Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.
GCredit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

COMPONENTS

Components for projects appearing in Elektor are usually available from certain advertisers in this magazine. If difficulties in the supply of components are envisaged, a source will normally be advised in the article. Note, however, that the source(s) given is (are) not exclusive.

TERMS OF BUSINESS

Delivery Although every effort will be made to dispatch your order within 2-3 weeks from receipt of your instructions, we can not guarantee this time scale for all orders. **Returns** Faulty goods or goods sent in error may be returned for replacement or refund, but not before obtaining our consent. All goods returned should be packed securely in a padded bag or box, enclosing a covering letter stating the dispatch note number. If the goods are returned because of a mistake on our part, we will refund the return postage. **Damaged goods** Claims for damaged goods must be received at our Brentford office within 10-days (UK); 14-days (Europe) or 21-days (all other countries). **Cancelled orders** All cancelled orders will be subject to a 10% handling charge with a minimum charge of £5.00. **Patents** Patent protection may exist in respect of circuits, devices, components, and so on, described in our books and magazines. Elektor does not accept responsibility or liability for failing to identify such patent or other protection. **Copyright** All drawings, photographs, articles, printed circuit boards, programmed integrated circuits, diskettes and software carriers published in our books and magazines (other than in third-party advertisements) are copyright and may not be reproduced or transmitted in any form or by any means, including photocopying and recording, in whole or in part, without the prior permission of Elektor in writing. Such written permission must also be obtained before any part of these publications is stored in a retrieval system of any nature. Notwithstanding the above, printed-circuit boards may be produced for private and personal use without prior permission. **Limitation of liability** Elektor shall not be liable in contract, tort, or otherwise, for any loss or damage suffered by the purchaser whatsoever or howsoever arising out of, or in connexion with, the supply of goods or services by Elektor other than to supply goods as described or, at the option of Elektor, to refund the purchaser any money paid in respect of the goods. **Law** Any question relating to the supply of goods and services by Elektor shall be determined in all respects by the laws of England.

January 2011

SUBSCRIPTION RATES FOR ANNUAL SUBSCRIPTION

	Standard	Plus
United Kingdom & Ireland	£51.00	£63.50
Surface Mail Rest of the World	£65.00	£77.50
Airmail Rest of the World	£82.00	£94.50
USA & Canada	See www.elektor.com/usa for special offers	

HOW TO PAY

Bank transfer into account no. 40270211 held by Elektor International Media BV with The Royal Bank of Scotland, London. IBAN: GB96 ABNA 4050 3040 2702 11. BIC: ABNAGB2L. Currency: sterling (UKP). Please ensure your full name and address gets communicated to us.
Cheque sent by post, made payable to Elektor Electronics. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.
Credit card VISA and MasterCard can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

SUBSCRIPTION CONDITIONS

The standard subscription order period is twelve months. If a permanent change of address during the subscription period means that copies have to be despatched by a more expensive service, no extra charge will be made. Conversely, no refund will be made, nor expiry date extended, if a change of address allows the use of a cheaper service.

Student applications, which qualify for a 20% (twenty per cent) reduction in current rates, must be supported by evidence of studentship signed by the head of the college, school or university faculty.
A standard Student Subscription costs £40.80, a Student Subscription-Plus costs £53.30 (UK only).

Please note that new subscriptions take about four weeks from receipt of order to become effective.
Cancelled subscriptions will be subject to a charge of 25% (twenty-five per cent) of the full subscription price or £7.50, whichever is the higher, plus the cost of any issues already dispatched. Subsciptions cannot be cancelled after they have run for six months or more.

January 2011

Elektor OSPV

→ **Open Source Personal Vehicle**

Last year we launched the Elektor Wheelie, a self-balancing personal transport device. Our new Elektor OSPV is based on the same concept, but with the difference that it's for indoors, it's easy to steer, it's light and foldable and… it's open source. You can configure or modify it to suit your wishes! The OSPV is primarily intended for moving people, but it doesn't have to be limited to that. A variety of other uses are conceivable, ranging from an electric wheelbarrow to a handy motorised shopping cart. This is where the advantages of the open source approach come to the fore!

elektor
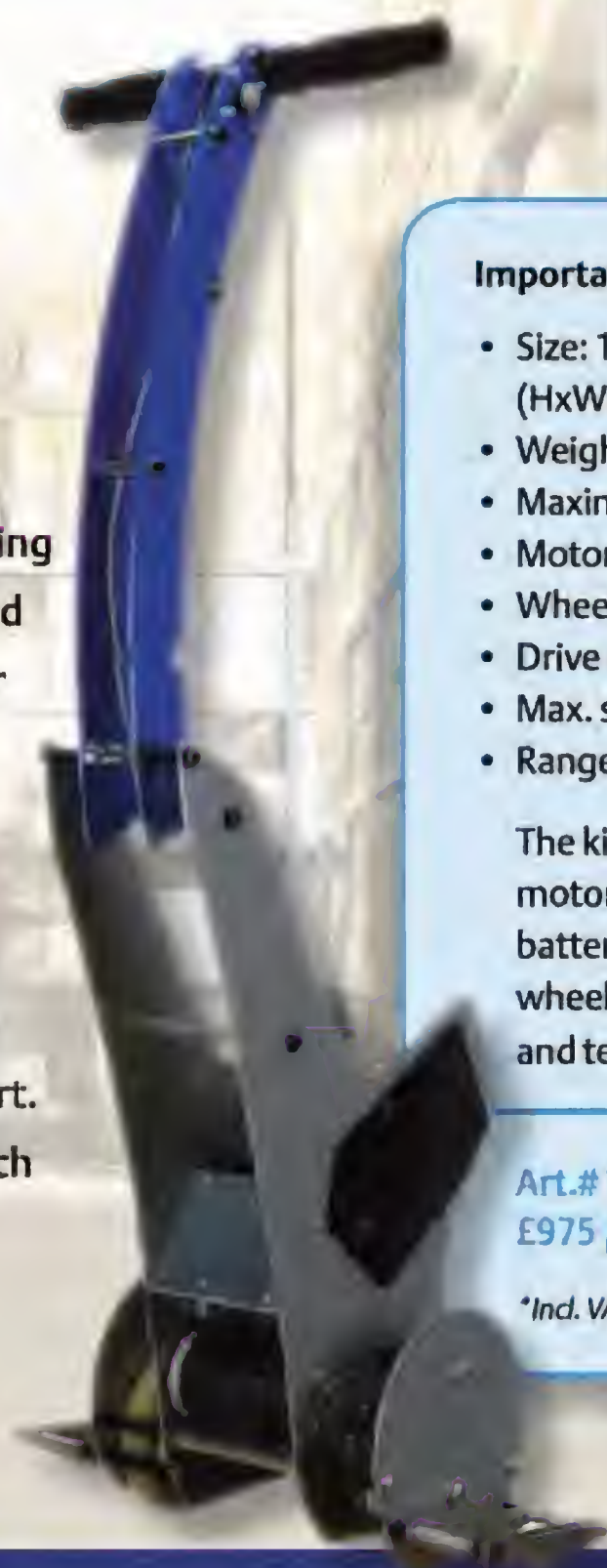
Important specifications:

- Size: 120x47x47 cm / 47.2x18.5x18.5 inch (HxWxD)
- Weight: 25 kg (25 lbs)
- Maximum load: 90 kg (200 lbs)
- Motors: DC 2 x 200 W
- Wheels: Polyurethane, 14 cm dia. (5.5 inch)
- Drive train: HDT toothed belt
- Max. speed: 15 km/h (9.3 mph)
- Range: 8 km (5 miles)

The kit comprises two 200-watt DC drive motors, two 12-V lead-acid AGM batteries, battery charger, two wheels Polyurethane 14 cm wheels, casing, control lever and fully assembled and tested control board with sensor board.

Art.#110320-91
£975 / € 1095 / US $1570

Incl. VAT, exd. shipping costs

Further information and ordering at www.elektor.com/ospv

INDEX OF ADVERTISERS